LINUX MAGAZINE

THE CHRONICLE OF THE REVOLUTION

NETWORKING:

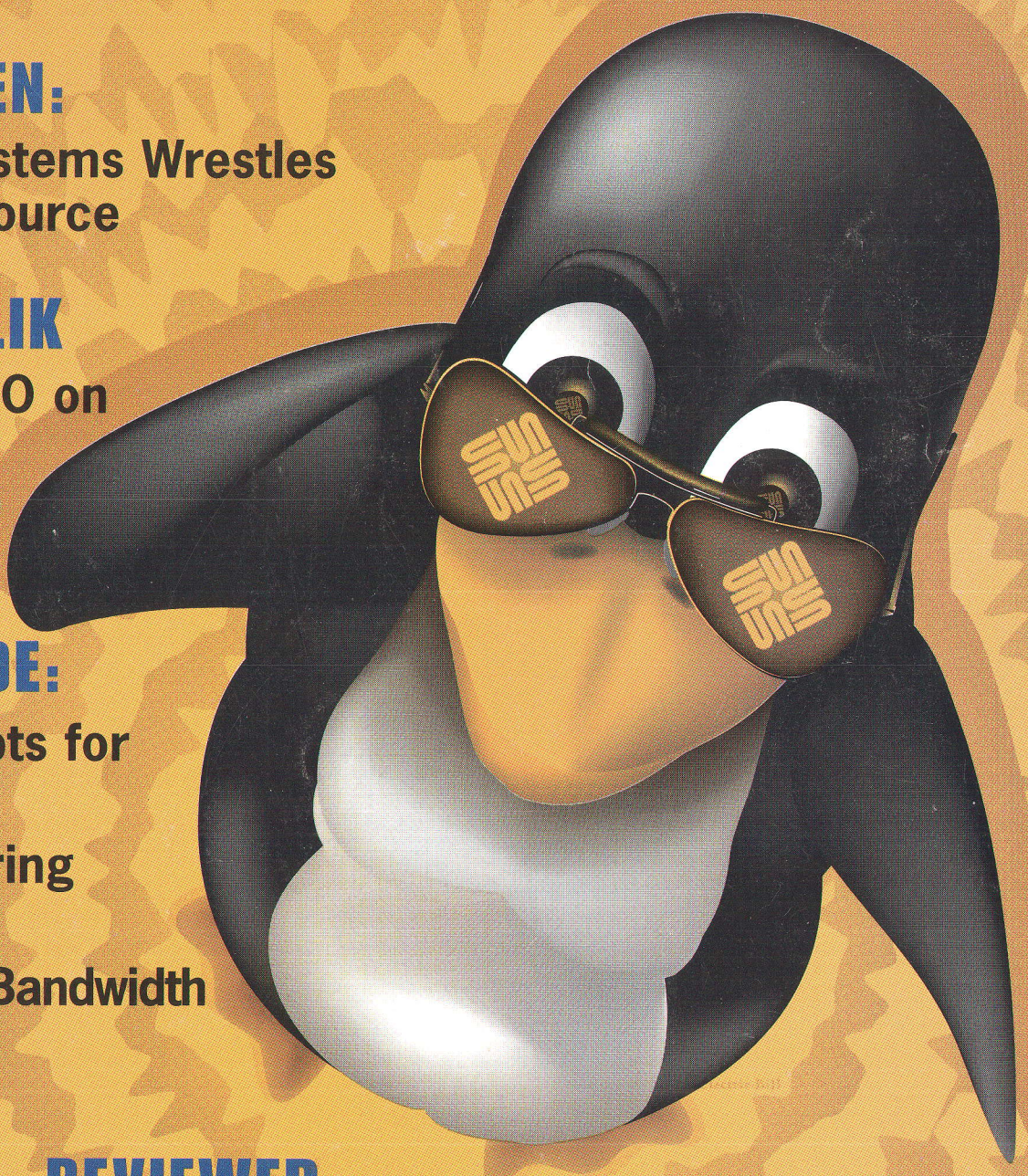
- Inside SAMBA
- Internet Routing With RIP and RIPv2

SUN SCREEN:

Sun Microsystems Wrestles With Open Source

MATT SZULIK

Red Hat's CEO on Building an Open Source Company



ALSO INSIDE:

- Shell Scripts for Newbies
- Administering E-mail
- Graphing Bandwidth Usage

MAY 2001

$4.95US $5.95CAN

05>

0 71486 01105 7

WWW.LINUX-

REVIEWED:

- Microlite BackupEDGE
- Compaq ProLiant Servers
- GreatBridge PostgreSQL

Other network attached storage includes something ours doesn't.

Namely, a big fat charge for software.

Here are your choices. The other guys' NAS that includes a big fat charge for their proprietary software. Or our high performance, high density VA Linux 9450 NAS. It comes with, for no charge whatsoever, our comprehensive Open Source NetAttach software, which includes a Logical Volume Manager, snapshot capability,* journalled file system, remote web-based management and more. The 9450 offers three PCI buses. Multiple RAID controllers. Integrated 10/100 or Gigabit Ethernet. And it scales from 180GB to 6.6TB in a 26U rack.** So your best NAS choice? Wave goodbye to the other guys. Call 1-877-VA-LINUX, or visit www.valinux.com. And get more NAS for less.

VA LINUX SYSTEMS

The World's Linux Leader.

My data center stays up all night. I don't have to.

Linuxcare and Turbolinux have joined forces to deliver the most complete high-availability solutions for your data center.

Now one company offers the most powerful clustering and supercomputing software—backed by the best services in the industry. Turbocharge your server farm or data center with scalable, reliable, open-source-powered solutions that perform no matter which Linux distribution you prefer. Find out more at www.turbolinux.com today.

turbolinux

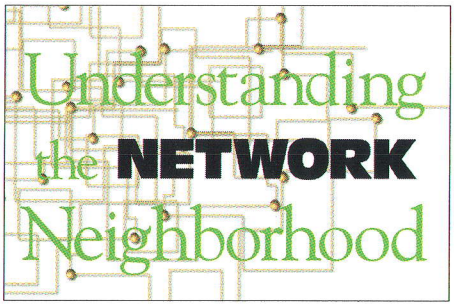
LINUXCARE

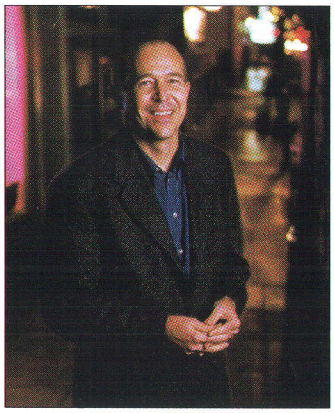
Powerful Thinking^2

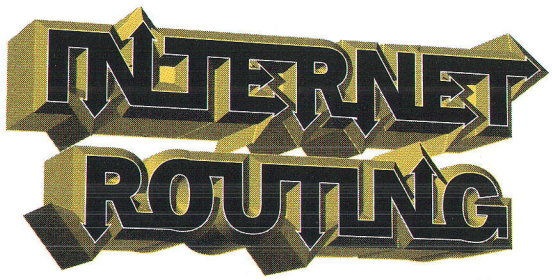
CONTENTS

MAY 2001

FEATURES







LINUX MAGAZINE
THE CHRONICLE OF THE REVOLUTION

CONTENTS

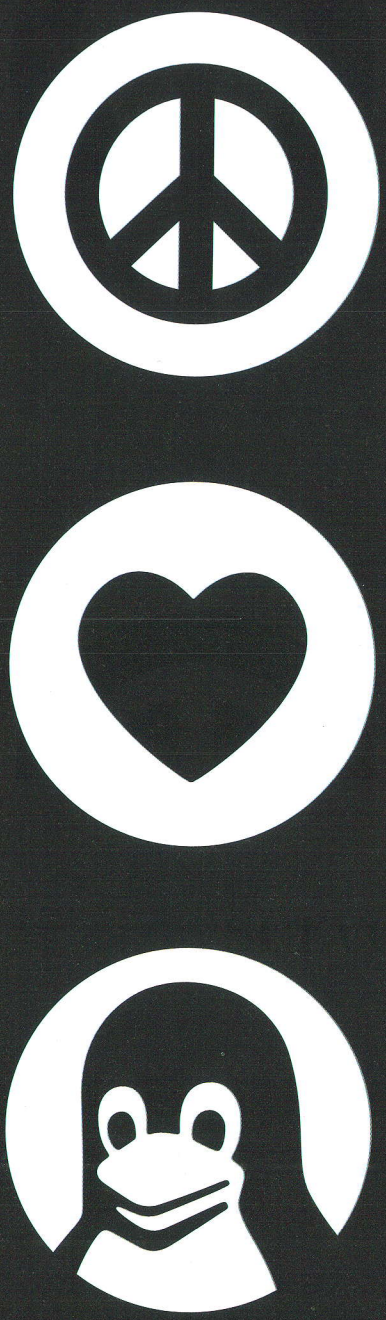
MAY 2001

DEPARTMENTS

REPORT FROM THE FRONT 10

REVIEWS 12

ZONKER'S PRODUCT PICKS 84



IBM | @server

Services, Baby, Services

Services. In the past few years, as Linux has developed into a more mainstream operating system, a new axiom has emerged — you can't make much money selling the OS itself, but many companies can make money by selling their services around the OS.

About a year ago, Tim O'Reilly wrote what I believe to be an incredibly insightful piece on this subject (found online at http://www.oreillynet.com/pub/a/linux/2000/05/09/lessons.html). I think that it is now worthwhile to re-visit this issue and take a close look at how some of the more promising open source business models have been evolving.

The first example that leaps to mind is Red Hat. While Red Hat has derived a great deal of their revenue to date from traditional "service and support" contracts with large corporate customers, it's clear that they are continuing to look for new models that will support their business. The most interesting of these new models is the Red Hat Network (RHN).

The RHN is based on a subscription model, where users of Red Hat Linux register themselves with RHN. The Red Hat Network will then provide users with constantly updated packages and bug fixes for all of the software that is installed on their systems.

Basic service is free for all users of Red Hat 6.2 or greater. However, many useful premium services are only available for a fee — on a subscription basis. So basically, Red Hat is helping their customers by managing their software for them over the Internet. If users find this to be a valuable service (and I certainly believe they will), Red Hat will derive a large revenue stream from it.

Hmmm...The two key words here seem to be "managing" and "service." This leads me to the next business model that I have seen emerging over the last year or so; there are now a slew of companies that have sprung up to offer managed hosting services to corporate America.

From traditional hardware manufacturers such as Dell (DellHost), Micron (HostPro), and IBM to new upstart "pure play" service providers such as Rackspace and Verio, an enormous number of service providers are offering corporate IT departments the option of outsourcing their entire infrastructure.

All of these providers offer their customers the opportunity to save time and money by having someone else deal with the headaches of managing their systems. Many of these providers do, in fact, use Linux as the foundation of their service offerings.

So, is it possible for you to make money utilizing the traditional "customer service and support" model and Linux? Absolutely. IBM Global Services, for example, is quite likely to make a mint. Many distributions and traditional support companies stand a rather good chance of capturing some of that business as well — not to mention the army of VARs that I believe will want to begin selling Linux to their clients.

The point I'm trying to make is that (as Tim O'Reilly also noted) opportunities here are much greater than what exists in the traditional model alone. Linux and the Internet together are causing dislocation in the world of information technology. Those companies that can identify these dislocations and use them to provide customers with better, cheaper, and more efficient ways of running their businesses will be the ones that are likely to benefit the most.

See you next month,

Adam M. Goodman
President & Publisher

LINUX MAGAZINE
THE CHRONICLE OF THE REVOLUTION

Editor & Publisher
Adam M. Goodman

Associate Publisher
Lara J. Kisielewska

Managing Editor
Christopher Somerville

Senior Editors
Pete Comas, Matt Welsh, Jeremy Zawodny

Editor at Large
Robert McMillan

Reviews Editor
Ross M. Greenberg

Contributing Editors
Joe "Zonker" Brockmeier, Æleen Frisch, Bill McCarty, Jason Perlow, Kevin Railsback, Paul "Rusty" Russell, Randal L. Schwartz, Bill von Hagen

Copy Editor
Jason Brainard

Legal Editor
Steven D. Lavine

Art Director
Rebecca E. Blake

Graphic Designers
Victor J. Brunetti, Ginny Chu, L. Jeanne Veronica

Production Manager
Meredith Rubin

Prepress Technicians
Alberto Lazo, Patty Nohara

Marketing Director
Mike Schaps

National Sales Manager
Eric Derby

Advertising Coordinators
Larissa Jeveki, Nancy Prestia

Circulation Director
Melissa A. Witham

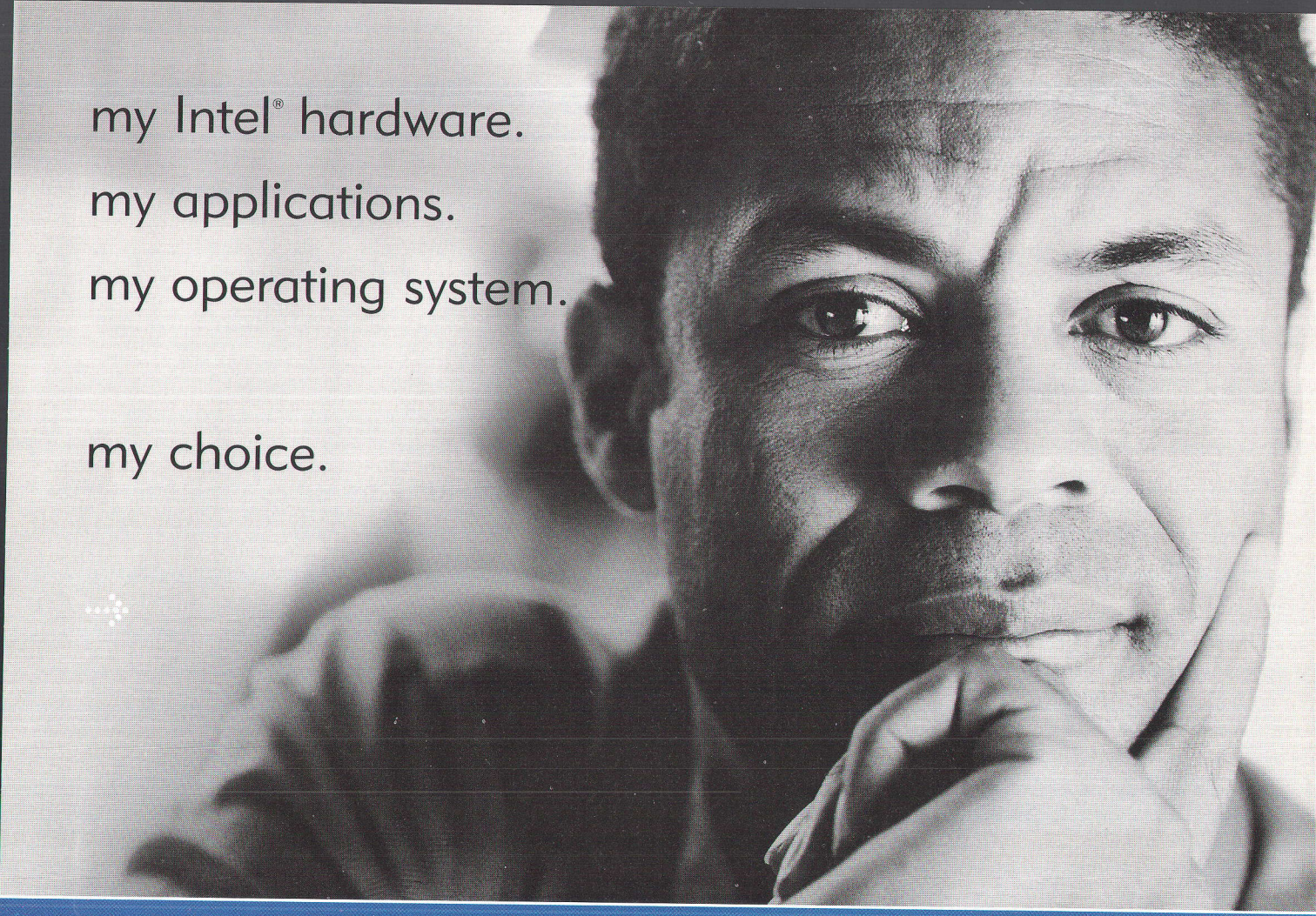
Newsstand Distribution
Irwin E. Billman, Ralph J. Perricelli

Marketing Assistants
Greg Aparicio, Paul D. Litt

Web Master
Jon Udell

Accounting Services
Jerome Molejona, Justin Stuhaan

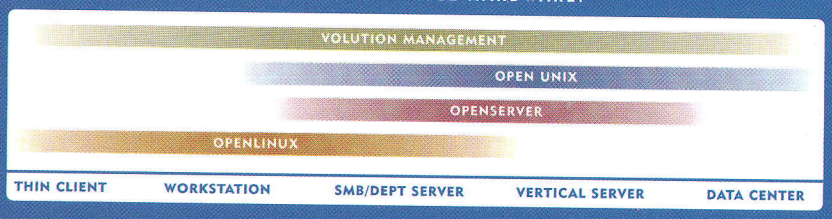
BPA Membership Applied For July 2000



Finally, IT professionals everywhere have an alternative.

Caldera® is harnessing the power of the Web through Linux®, and the stability and scalability of the back office through UNIX®, to give you a single, cohesive family of solutions for your Intel hardware. From thin client to data center, now you have a powerful software choice – Caldera.

CALDERA'S SOFTWARE STRATEGY FOR INTEL HARDWARE:



Visit **www.caldera.com/offer/4** for software strategy information, and get the FREE white paper: **"Caldera gives you a choice"**

CALDERA systems | SCO

CALDERA®

www.caldera.com

A NEW COMPANY FOR A NEW ERA

—

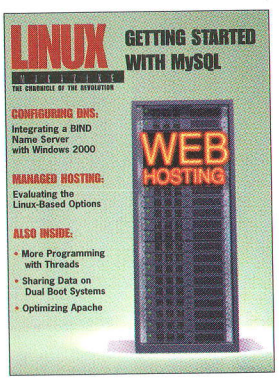
UNIFYING UNIX® WITH LINUX® FOR BUSINESS™

Some Major Partners Include: Compaq®, Intel®, Computer Associates®, IBM®

Caldera has over 20 offices in 80 countries, and world-class education, professional services, and support.

Beware Software Giants Bearing Gifts

I just finished reading "Embrace & Extend" (February 2001) and believe that your position regarding Microsoft is a bit naive. However well-reasoned Microsoft's .NET may be — and I agree it *sounds* good — we cannot afford the luxury of ignoring the Microsoft common denominator; everything Microsoft does with regard to the Internet is a clandestine attempt to gain proprietary control of the Internet or an indirect attempt to control what software people use. Evidence to support this is so overwhelming that only Microsoft's lawyers think there is any point in discussing it.



The bottom line is that we cannot afford to use anything from Redmond, no matter how appealing and seemingly useful. If just one proprietary Microsoft tool becomes an Internet standard, we will have made a gift of the whole Internet to Uncle Bill. Be assured, there will be something attached to the seemingly free gift. I would rather spend the rest of my life chopping wood with a stone axe than run the risk of buying an electric heater from Redmond. The price is too high.

At the moment, total control of the Internet is in the hands of Unix/Linux. Gates knows this and wants to change it. Not one brick used to construct the Internet must ever come from Redmond. .NET is a Trojan horse if ever there was one.

David Kachel
The Mac Workshop

My Compliments to the Chef

I just got through reading "The Truth About Text — Part III" (*Newbies*, March 2001). Great article! I learned a lot about *vi*. Can I look forward to an article or two on *emacs*? I've always been mystified by *vi;* I know it's a powerful text editor, and your articles did much to help me realize why it's on nearly every Unix/Linux system. I know *emacs* is also quite powerful and would like to learn how it works and what makes it so great.

I also appreciated "Hey! Leggo MySQL" (March 2001). Hopefully, there will be more articles on MySQL in the near future. Of course, I'd also like to find one on PostgreSQL, since I know that there is "competition" between these two database apps.

Thank you for your magazine; it keeps me up-to-date and gives good information in a pleasant, informative (and occasionally humorous) way.

Russell Hires

Easier than procmail

You recently responded to a *Tech Support* question (March 2001) about archiving inbound e-mail and suggested using *procmail*. A simpler solution that works on almost all Unix systems is to use the standard *.forward* file capability of *sendmail* and *sendmail*-compatible transport agents. The *.forward* file looks like:

```
\username, /home/username/
    archive/email
```

The backslash in front of the username prevents further aliasing or forwarding. The second entry is the name of a file to which *sendmail* will write the incoming mail.

While obviously not as flexible as the pattern matching capabilities of *procmail, sendmail* does the job and is much easier for the user to configure if *procmail* is not available on their system.

Dan Trottier

Check out this month's Guru Guidance *column for more on this subject. –Ed*

Windows and Linux Coexisting

The *Tech Support* column in your March 2001 issue really caught my eye. Unfortunately, it only addressed how to mount and access data on the Windows partition. What would be really useful is an in-depth article discussing how data can be shared between applications in Windows and Linux.

For example, is it possible to set up Netscape to use the same bookmark file in Windows and Linux? How about mail files? What are some good practices for setting up a directory structure to support file sharing?

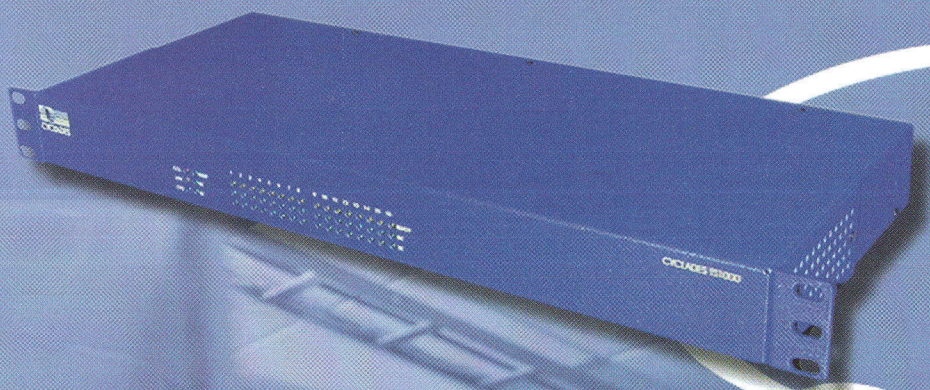
While I find myself using Linux more, there are still some applications I really want to use that are only available under Windows — in particular, my Palm conduit applications. While I've set up J-Pilot and can do Hot Syncs, there is still an absence of conduits and desktop applications that utilize data on the Palm.

Pocket Quicken from Landsware, for example, allows you to sync financial data on the Palm with the Quicken application. GnuCash doesn't have such a conduit yet. It is these types of applications that will help convince me to move to a single boot system. Until then, I plan to use both OSes. Any information on sharing data between the two would be a welcomed addition to your fine magazine.

Tom Newman
Clarke County, VA

Linux Magazine *welcomes feedback and comments. Letters will be edited for clarity and space; please be sure to include your name and location. Send letters to editors@linux-mag.com.*

Console access & Terminal server



if only everything was this easy.

TS2000

The TS2000 provides secure access to the console ports of servers and networking equipment. Now you can easily manage and troubleshoot your network from anywhere, anytime.

- **32 RS-232 ports on 1U of rack space.**
- **First Linux-based Terminal Server in the market.**
- **Packet Filtering, RADIUS, and Secure Shell (SSH).**
- **Status LEDs on the front panel for all ports.**

Also available:
TS1000 (16 ports)

Hard Hat Linux Inside

1995 1996 1997 1998 1999 2000 Readers Choice

THE LEADER IN LINUX CONNECTIVITY

www.cyclades.com

1.888.CYCLADES 1.888.292.5233
510.770.9727
sales@cyclades.com
Fremont, CA

CYCLADES

Report from the Front

Microsoft VP Blasts GPL

The war of words between Microsoft and Free Software and Open Source proponents heated up in February. Jim Alchin, VP of Microsoft's Platforms Group, called Open Source an "intellectual property destroyer" in a February interview. Alchin's comments were specifically targeted at legislators; He claimed that GPLed or Open Source software stifles innovation by companies like Microsoft. Alchin's comments came about a month after Microsoft's Steve Ballmer called Linux their top threat.

Microsoft officials later clarified that Alchin's primary concern is the fact that the GNU General Public License stipulates that code licensed under the GPL must stay under the GPL if modified. Apparently, Microsoft feels that BSD-style licenses that do not require third parties to give back code are fine. http://www.microsoft.com

IBM Adopts Linux for New Storage Standard

IBM is developing a new standard for Network Attached Storage (NAS). iSCSI is based on the SCSI protocol but operates over TCP/IP networks.

The TotalStorage IP 200i is a Linux-based NAS solution that implements the iSCSI protocol. IBM plans to introduce the IP 100i later this year with prices starting at $20,000.

The iSCSI protocol is part of a push to deliver an open network storage solution. IBM is working with Cisco Systems to make it an Internet Engineering Task Force (IETF) standard. http://www.ibm.com

Hewlett-Packard Ends OpenMail

Hewlett-Packard recently announced that the upcoming 7.0 release of their OpenMail groupware messaging product would in fact be its final major release.

OpenMail is the only messaging software currently available for Unix and Linux that supports Microsoft's MAPI protocol. This allows companies to take advantage of some of Microsoft Outlook's capabilities without being forced to standardize their systems on Microsoft Exchange.

Bruce Perens, HP's Senior Strategist for Linux and Open Source, indicated that there are discussions underway within HP that may lead to OpenMail being released under an Open Source license, but no firm decision has yet been reached.

HP announced that they will support OpenMail 6.0 and 7.0 for five years, but that they will not be developing new versions or a replacement package. http://www.hp.com

Linux Breaking into Retail

Linux is set to realize a 300 to 400 percent growth in the retail point-of-sale (POS) market this year, according to the IHL Consulting Group.

In 2000, Linux had but a 2 percent share of the POS market, but with large rollouts from Home Depot and MusicLand totaling approximately 25,000 Linux units, 2001 looks to be the year that the open source OS makes it big in retail.

According to IHL's study, large retailers and specialty retailers with larger IT staffs are more likely to embrace Linux as a POS solution. While Linux may not carry a license fee, smaller retailers might not be able to justify the overhead costs of implementing a Linux solution.

The study was conducted for the Association of Retail Technology Standards (ARTS). http://www.ihlservices.com

ActiveState to Support Tcl

The Tool Command Language, better known as Tcl (and usually found in conjunction with it's partner toolkit, Tk), has found a corporate supporter in ActiveState. Not only will the company be adding the language to its set of commercially supported tools, they will soon also agree to host the primary Tcl community Web site.

Tcl/Tk had been supported by Scriptics, a company that focused on Tcl/Tk solutions. However, in May of last year, Scriptics changed its name to Ajuba Solutions and its focus to XML-based business-to-business solutions.

Tcl is a popular scripting language in the open source community, while Tk is a toolkit that allows programmers to create GUI applications with Tcl. ActiveState provides tools and support for open source programming languages, including Perl and Python (and now Tcl), on a range of platforms. http://www.activestate.com

Sharp Linux PDA to Debut in October

The ever-elusive Linux PDA is likely to become a reality this October. Sharp has announced that it will introduce a Zaurus PDA model based on the Linux OS in Europe and the US during the fourth quarter of this year.

Sharp's Mobile Systems Division Chief, Hiroshi Uno, is optimistic about the device's success and says that the company intends to sell approximately one million units by the following March and eventually grab 50 percent of the PDA market.

According to Gartner, PDA sales will skyrocket between now and 2004, and Sharp is betting its share of the market on Linux and the large number of developers that are behind the Free Software OS.

While there are several free distributions and ports of Linux available that will run on a variety of PDA hardware, no commercial Linux-based PDA solutions exist currently. http://www.sharp-usa.com

Red Hat Buys Planning Technologies

In the latest of a series of acquisitions, Red Hat has announced the purchase of Planning Technologies, Inc. for $47 million in stock. With this acquisition, Red Hat adds to its payroll approximately 200 engineers from the Atlanta-based Planning Technologies.

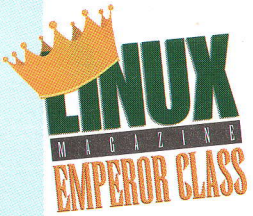
Planning Technologies specializes in infrastructure consulting for enterprise clients, service providers, and governmental agencies. The company was founded in 1992, and its client list includes AT&T and Delta Air Lines. Interestingly, PTI is not solely a Linux, or even open source, company; their focus has historically been on developing and deploying network architecture. http://www.redhat.com

Goin' Mobile: Transmeta Set to Release Mobile Linux

Transmeta has been working on more projects than just their Crusoe processor and Code Morphing software. They have also developed their own Linux distribution called Mobile Linux, which works with the Crusoe and Intel-based processors and is designed for use with Internet appliances such as Gateway's AOL Webpad.

Mobile Linux was designed specifically to be used with diskless appliances and therefore has lower memory requirements than Linux distributions that were designed for PCs and servers. Mobile Linux also emphasizes power conservation and management.

Of course, Transmeta has a not-so-secret weapon when it comes to developing new versions of Linux; they employ Linux creator Linus Torvalds. http://www.transmeta.com

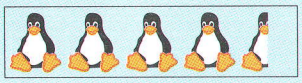


GreatBridge PostgreSQL

GreatBridge PostgreSQL 7.0.3 $Free
http://www.greatbridge.com/

In a Nutshell

Rating:

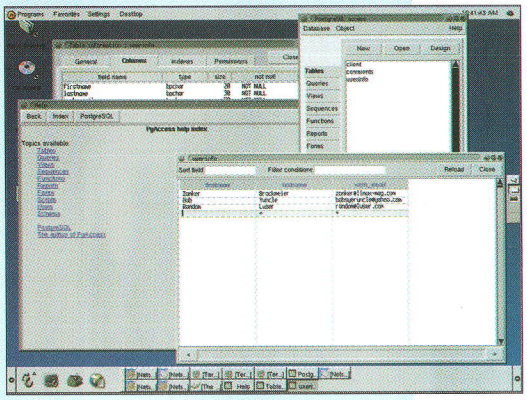


Pro:

- Great GUI client

Con:

- No Debian packages



Full Toolbox: PostgreSQL comes with many GUI tools.

Supported Platforms

- Red Hat 6.1, 6.2, 7.0
- SuSE 6.4, 7.0
- Turbolinux 6.0.4
- Mandrake 7.2
- Caldera eServer 2.3

System Requirements

Hard Disk

- 21 MB free space

Package Manager

- RPM

Available Resources

Downloads

- Distribution-specific RPMs
- Distribution-specific source code

Tech Support

- Available separately (call for pricing)

By Joe "Zonker" Brockmeier

How much would you expect to pay for an easy-to-install, full-featured SQL-compliant database? How about nothing? GreatBridge PostgreSQL is a great database package with all the bells and whistles that come in its high-priced brethren — but it's open source.

Both RPMs and source code are available for download, as well as an ISO image that comes complete with an installer. The distribution-specific ISO images are about 42 MB compressed, so they won't take long to download with a reasonably fast connection. Additionally, there is an option to sign up for a free GreatBridge PostgreSQL CD.

We installed GreatBridge PostgreSQL from the RPMs onto a SuSE 7.0 system. The installation was painless, and PostgreSQL was up and running in less than twenty minutes (not including the time it took to download the RPMs and read the install instructions).

Manuals and Tools

The accompanying documentation is complete and is available both online and in printable PDF format. There are three manuals available on the site: an Installation Guide, a User's Guide, and a Reference Guide. Unless you're already a PostgreSQL expert, you'll probably need all three.

PostgreSQL also comes with a user-friendly GUI client called PgAccess. PgAccess is a full-featured Tcl/Tk client that makes it easier to work with the database by handling everything from creating databases to adding users to creating queries. PgAccess allows you to create Tcl/Tk forms for data entry.

If you're more comfortable using SQL in a terminal environment, don't worry. The *psql* program lets you interact with the database from a terminal-based front-end. We spent a little while with *psql* and decided that PgAccess is more fun. Hardcore SQL maniacs may prefer *psql*, but PgAccess spoiled us.

PostgreSQL supports both TCP/IP and SSL connections. The Postgres *postmaster* program has to be explicitly told to allow connections over TCP/IP, otherwise it assumes you only want to be able to connect through domain sockets from the local machine. The first time we started PostgreSQL, we neglected to set these options and had to kill and restart the *postmaster* program.

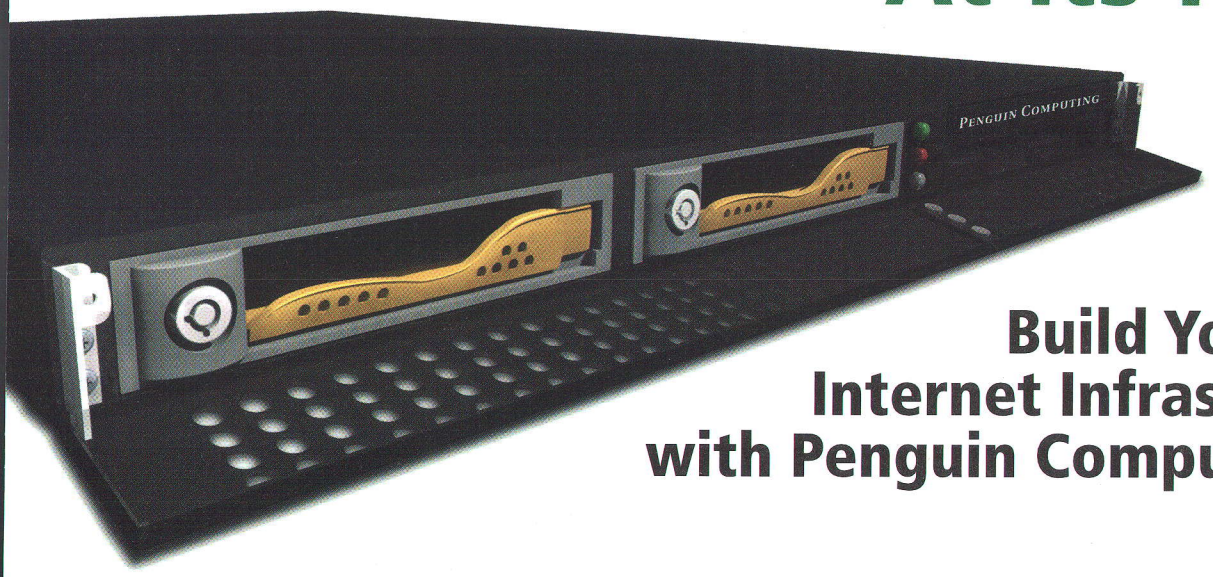
Minor Drawback

The only complaint we have about GreatBridge PostgreSQL is its lack of a distribution-agnostic download. While they do have tailored RPMs for the latest versions of the major RPM-based distros, if you're using a distribution that doesn't support RPMs, you may be out of luck. However, GreatBridge is only one of the organizations that works with PostgreSQL, so if you find yourself in this unlucky situation, you can still find vanilla source code for PostgreSQL elsewhere on the Internet.

GreatBridge doesn't make its money selling licenses to PostgreSQL or even by selling the CDs. It does ask for a little personal information when you download documentation, but there's no charge for the PostgreSQL software itself. Instead, the company has chosen to make its money from technical support, consulting, and other services.

GreatBridge PostgreSQL is a great package. If you're looking for an SQL-compliant database for your Web site or any other project, you should download GreatBridge PostgreSQL and try it out. It won't cost you a thing. LM

Internet Serving At Its Finest



Build Your Internet Infrastructure with Penguin Computing™

SINGLE CPU WEB SERVER
Relion 100E — Starting At **$1145**

- Slim 1U (1.75") form factor
- Intel® Celeron™ or Pentium® III processor options
- Up to 512 MB of PC100 memory (2 DIMM slots)
- Single integrated Ethernet port
- Integrated 2-channel EIDE controller
- Two low profile EIDE drive bays

DUAL CPU WEB SERVER
Relion 120 — Starting At **$2417**

- Slim 1U (1.75") form factor
- Single or dual Intel® Pentium® III processors up to 933 MHz
- Up to 4 GB of PC133 memory (4 DIMM slots)
- Dual integrated Ethernet ports
- Integrated Ultra160 SCSI controller
- Two low profile hot-swap SCA LVD drive bays

APPLICATION SERVER
Relion 220 — Starting At **$2293**

- 2U (3.5") form factor
- Single or dual Intel® Pentium® III processors up to 1.0 GHz
- Up to 4 GB of PC133 Memory (4 DIMM Slots)
- Dual integrated Ethernet ports
- Dual hot-swap power supplies available
- Six low profile hot-swap SCA LVD drive bays

WORKSTATION
Award-Winning Niveus — Starting At **$1183**

- Fully Integrated Linux Workstation
- Dual Intel® Pentium® III processors or single AMD Athlon™ processor
- Up to 1.5 GB of PC100 / 133 memory
- Highly scalable with up to six 5.25" drive bays
- Wide array of multimedia devices and upgrades

PENGUIN COMPUTING™

Buy Online: www.penguincomputing.com
Or Call: 1-888-PENGUIN

Tel: 415.358.2600 Fax: 415.358.2646 sales@penguincomputing.com

REVIEWS

Microlite BackupEDGE 7 $300
http://www.microlite.com

In a Nutshell

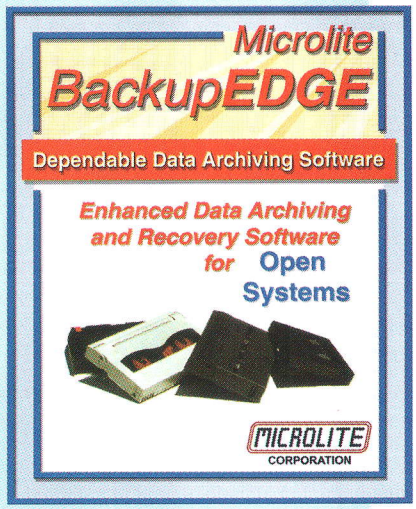
Rating: 

Pros:

- Extended SCSI support provides network device backups
- Master backups can be restarted
- Creates crash recovery diskettes

Cons:

- Backup format is proprietary
- No support for standard software compression (gzip), only internal software and hardware compression



Curses: Microlite provides old-school backup.

System Requirements

CPU

- 386 MHz or better
- 2.x kernel

Backup Drive

- SCSI or IDE tape drive

Supported Platforms

- AIX
- Linux
- Solaris
- Hewlett-Packard
- SCO Unix
- UnixWare

No Real BackupEDGE Here

By Bill von Hagen

Backups are a pain. They're one of those odious and time-consuming necessities of responsibly managing a computer system. If you do not do backups with some regularity, there will be nothing but old data around when a drive goes bad, and then you'll be screwed.

That's where BackupEDGE from Microlite Corp. comes into the picture. Microlite's serious, long-term experience on Unix platforms is obvious from the start; the "programs" that report on a backup's overall success or failure are shell scripts, making it easy to integrate custom reporting and site-specific error code handlers.

Backing Up

According to its documentation, backups done by BackupEDGE combine, "...many of the best features of *tar, cpio,* and *sysadmin*..." Unfortunately, that statement leads to our biggest complaint with BackupEDGE; while it may combine the best features of those products, its actual backup data is compatible with *none* of them. This makes BackupEDGE useful only if you plan to do all of your archiving *and* restoring using BackupEDGE. A better, platform-independent solution would include a compatibility mode, where *tar* or *cpio* could restore backups.

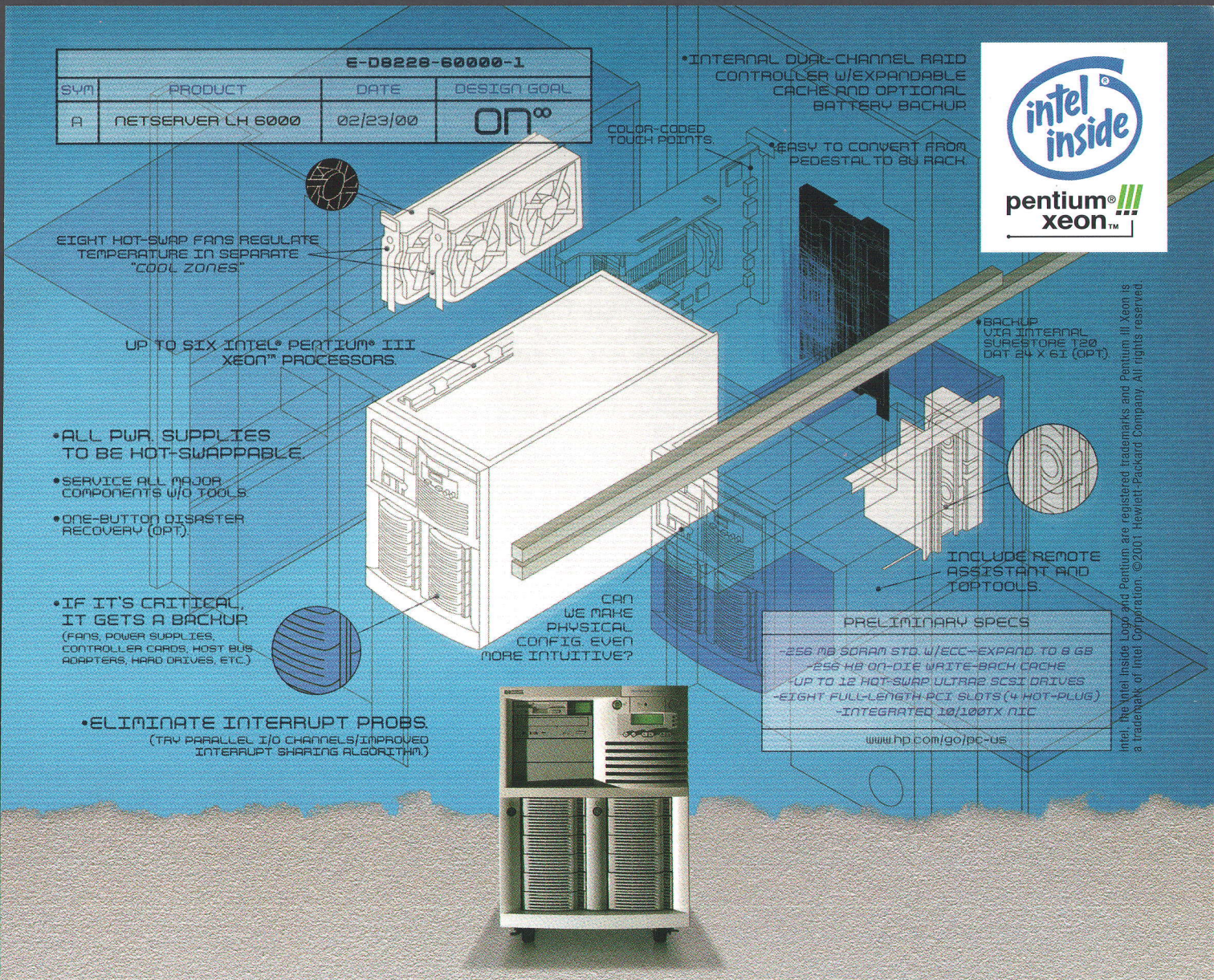
Backups performed by BackupEDGE do support both software and hardware compression (assuming that your backup device supports these as well), and both can be switched off if desired. A good solution for platform-independent software compression is to pipe the backup output into `gzip -9`. Perhaps this will be implemented in a future release.

In the interest of being truly platform independent, it would be nice if BackupEDGE allowed for backups to any target filesystems, even across a network. This would allow backups performed with BackupEDGE on one system to be restored on another. Unfortunately, this is not possible at this time. Currently, the only way to accomplish this is to move all of your backup devices to the target system, install BackupEDGE in demo mode there, and use this combination to restore your files.

Some Problems

The program's user interface is primarily a curses-based GUI. However, some functionality (like the index-based "fast file restore") is also available in an X Window system version. We'd prefer to see the whole thing ported to the X Window system. Curses-based interfaces are definitely lean and mean, but they look sort of kludgy nowadays. We also had some problems getting BackupEDGE's installer to work correctly from within an *xterm* window. Arrow keys did not work deterministically in the installer (a serious problem for selecting options!) unless run from a command line outside of the X Window system.

In short, BackupEDGE is a fine backup and restore utility if you intend to do all of your backups and restores only on the same hardware. It supports standard Master and Incremental backups, and BackupEDGE mails a nice summary of its backup operations to a user-specified e-mail address. The program has an excellent track record on both Unix and Linux. However, if hardware and software compression isn't mandatory or available in your computing environment, then you might find that the standard Unix *tar-gzip-dump-restore* utilities contain most of the same functionality that you will find in BackupEDGE. LM



The real genius lies in what you do with it.

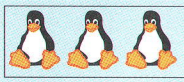
invent

Compaq's Slender Beauties

By Alan Zeichick

Compaq ProLiant DL320 Server
$2,541 Base, $4,434 as Tested
http://www.compaq.com/products/servers/proliant-dl320/index.html

In a Nutshell

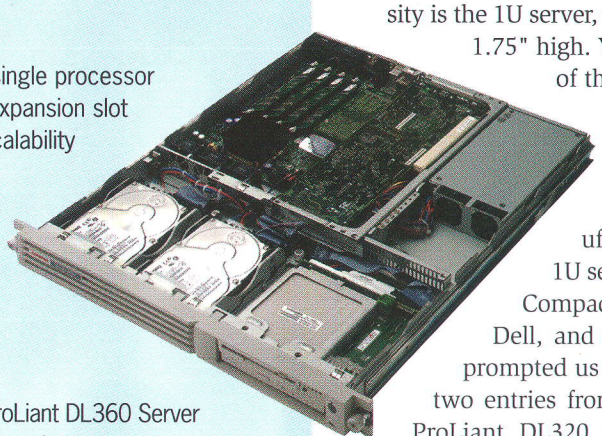
Rating: 

Pros:

- Takes up little room in server rack
- Inexpensive if purchased without a SCSI-based disk system

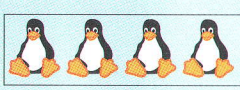
Cons:

- Only a single processor
- Single expansion slot limits scalability



Compaq ProLiant DL360 Server
$3,986 Base, $8,081 as Tested
http://www.compaq.com/products/servers/proliant-dl360/index.html

In a Nutshell

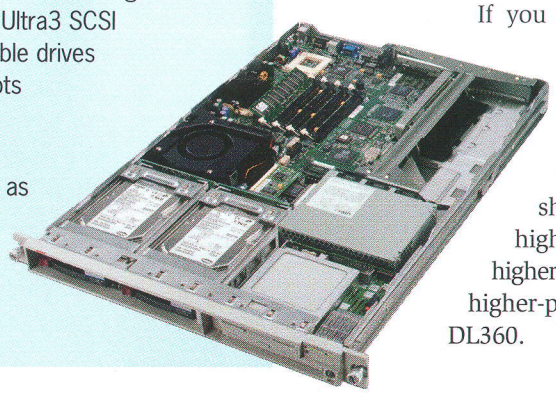
Rating: 

Pros:

- Takes up little room in server rack
- Lets you have up to 42 servers (84 processors!) in a single rack
- High-speed Ultra3 SCSI
- Hot-swappable drives
- Dual PCI slots

Con:

- Nearly twice as expensive as the DL320



When it comes to servers, you can choose one of two paths. You can go for *raw power*, with four-way or eight-way processors, gigabytes of RAM, dozens of hot-swap hard drives, and hot-swap PCI buses. Or you can go for *density*, stuffing as many servers into as small a space as possible.

The ultimate expression of rack density is the 1U server, which stands at 1.75" high. You can stuff 42 of these puppies inside a typical server rack. Most major hardware manufacturers offer 1U servers, including Compaq, Data General, Dell, and IBM, but what prompted us to review these two entries from Compaq, the ProLiant DL320 and DL360, is that the company offers two different models — they are outwardly quite similar but have distinct design philosophies.

During the review process, both models performed spectacularly under Red Hat 7.0, our test distribution. It's not a question of whether these are good servers for a high-density data center; the issue is which server fits the application best. If you're running a Web server or file server, you should go with the less expensive DL320.

If you are running an application server or perhaps a transactional database, then you should choose the higher-performance, higher-availability (and higher-priced) ProLiant DL360.

Compare and Contrast

The biggest differences between the servers can be found in two areas: processor and disk subsystem. The DL320 has a single Pentium III processor and offers either ATA (also known as IDE) or Ultra2 SCSI-based disks; the tested model had a single 800 MHz Pentium III processor, 1 GB RAM, and two 10 GB ATA/100 drives. The DL360, which looks outwardly quite similar, comes with one or two processors and Ultra3 SCSI-based disks; the evaluated system had dual 800 MHz Pentium III processors, 1 GB RAM, and two hot-swappable 18 GB, 10,000 RPM Ultra3 hard disks. (Both servers also come in faster versions — up to 933 MHz for the DL320 and 1 GHZ for the DL360.)

When you get beyond those important differences, the machines start to look the same. Both have dual 10/100 network interface cards, for example, which were immediately identified and properly configured by Red Hat 7. Both have lots of video RAM so that you can have a decent sized X Window console if you like administering a server via KDE or GNOME. Both are limited to a single power supply. Both have a wide range of OS support; Compaq officially supports not only Red Hat, but also SuSE 6.3 and 7.0, Turbolinux Server 6, and Caldera eServer 2.3. The company offers a preload of Red Hat 7, but we manually installed the OS onto both servers.

What are these servers going to set you back? As equipped for this review, the DL320 has a list price of $4,434. The DL360 costs $8,081 — nearly twice the price.

It's All In the Details

In many ways, the ProLiant DL320 is like a Compaq consumer PC squashed into a pizza box. With the single processor and dual-channel 100 Mbps ATA

disk bus, that's mainly what it is. Don't underestimate a single processor machine, particularly for lightweight tasks like file serving or simple Web serving (which isn't too different from file serving) or driving a sendmail system. On the other hand, some tasks — running an application server or Web site with Java servlets, a transaction database, acting as a Beowulf cluster, racking up a high score at SETI@home — should really have the dual-processor power of the DL360.

What about the disk I/O subsystem? For the past ten years, it's been "assumed" that a server uses SCSI hard drives and that desktops use ATA drives. That made a lot of sense. Back when drives were flakey, the ATA bus (which means "AT Attachment"; it used to be called IDE, for "Integrated Drive Electronics") poked along at a snail's pace. But the latest revision of the ATA spec, ATA/100, zips along at 100 Mbps — faster than Ultra2 SCSI's 80 Mbps. Given that only two hard drives can sit on an ATA bus (compared to 7 or 15 on a SCSI bus), a 100 Mbps bus is more than adequate.

Our only beef with the DL320's ATA system is that the hard disks themselves are slow. The DL320 we tested used the 10 GB Seagate Barracuda ST310216A drive, with a rotation speed of 7,200 RPM and 8.2 milisecond access time. By comparison, the 18 GB Ultra3 SCSI (160 Mbps) drives in the DL360 spin at 10,000 RPM and has a 5.2 milisecond access time. For driving a database, you're better off with the faster drives and the SCSI subsystem. But if the server is stuck in the Web server role, processing middleware, you should be happy with ATA.

Another difference in the servers' disk subsystems is that the DL360 includes a hardware-based RAID controller, which can either span the two disks into a single logical volume or set up the drives with hardware-based mirroring. If that's important, then the DL360 is the only game in town.

A final difference worth pointing out is in server expendability. The DL320 has only a single PCI 64-bit 33 MHz expansion slot — we couldn't see why; there's plenty of room inside the chassis to stuff a second slot. But if you like gigabit Ethernet, want to put a SCSI card inside to drive an external tape array, or want to put the server on a Fibre Channel-based storage area network, you'll be only able to pick one. By comparison, the DL360 has two 33 MHz slots (one 32-bit and one 64-bit) and also includes an external SCSI port for a tape drive.

Bottom Line

Both servers get the job done, though the DL360 is clearly a better machine, offering additional performance and scalability, albeit at a higher price. If your goal is to pack a lot of processors and OSes into a tight space, either server can do the job. With the DL320 for lighter-weight tasks, and the extra oomph provided by the dual-processor DL360, you'll have a good combination for most data-center tasks. LM

NEWBIES

Learning a Script

By Bill McCarty

It seems that one of the skills required of an actor is the ability to memorize a script. I had assumed that learning a script is of greater importance to stage actors than film actors, because stage actors must deliver their lines and follow their stage directions correctly the first time. However, acquaintances that are in the business assure me scripts play a central role in both types of acting.

A First Script

With all the copyright disputes that have arisen lately, I'm surprised nobody has commented on Burger King's ownership of the slogan "Have It Your Way." That's been the slogan of Linux and Unix developers since the beginning; if you don't like how your system works, write a script to make it work your way.

By constructing scripts that perform commonly used operations, you can reduce the tedium and effort these operations require.

Scripts of a somewhat different kind are important to Linux and Unix users. Scripts (sometimes referred to as "shell scripts," since the scripting language is built into the shell) let you teach a computer new commands of your own design. By constructing scripts that perform commonly used operations, you can reduce the tedium and effort these operations require. A thorough knowledge of scripting is one of the keystones of Linux mastery.

For instance, suppose you often use the *w* command to see which users are logged into a host (see *Figure One*).

It's difficult to see if a specific user is logged on, because the *w* command prints its output in no particular order. By writing a script, however, you can change this behavior. Using a text editor, create a file named *users* with the following contents:

```
w | sort
```

Now, execute the file by issuing the following command:

```
sh users
```

The output will now resemble *Figure Two* (pg. 20), in which lines appear in an order governed by the user name.

Okay, the heading line appears last in the output, but we'll fix that in a moment. First, let's look at how the script works. Essentially, typing the command `sh users` is no different than typing the command contained in the file `w | sort`. This command executes the *w* subcommand and then "pipes," or sends, the *w* command's output to the *sort* subcommand, which alphabetizes the *w* command's output.

You may reasonably object on the grounds that it's almost as convenient to type `w | sort` as it is to type `sh users`. Fair enough; let's fix that. Issue the following command:

```
chmod u+x users
```

Now you can obtain the sorted listing by merely typing the command `./users`. That's easier than typing `w`

Figure One: Output of the w Command

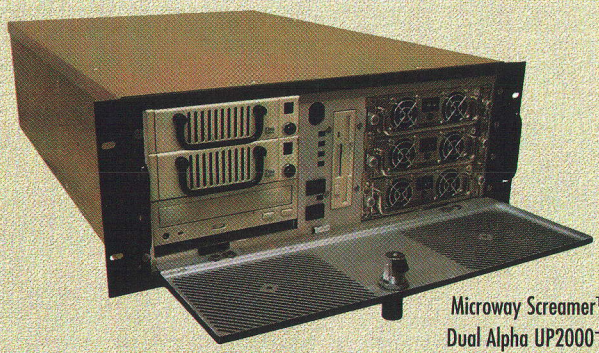
```
[bmccarty@home bmccarty]$ w
10:06am    up 149 days, 22:01, 38 users,  load average: 0.00, 0.00, 0.00
USER       TTY      FROM          LOGIN@   IDLE     JCPU      PCPU      WHAT
dglyer     pts/0    faedhcp117    Thu10am  23:54m   0.73s     0.01s     /usr/local/bin/pdme
sniper86   pts/25   -             10:05am  1.00s    0.07s     0.05s     vi .te/aliases.irc
enigma     pts/22   -             7:33am   35:18    60.41s    60.41s    irc -c#tech enigma
philpi     pts/31   -             9:13am   5:34     0.30s     0.30s     pine
mccartyp   pts/21   -             9:55am   11:02    48.52s    48.51s    pine -i
enigma     pts/1    -             7:33am   35:40    1.16s     1.13s     irc -c#linux enigma

etc.
```

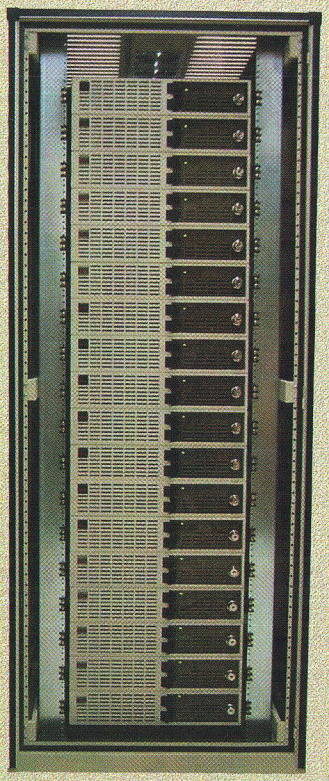
Microway Delivers Custom Linux Workstations and Beowulf Clusters for Linux Professionals!

Which Solution is Right for You?

- 833 MHz Dual Alpha
- 500 MHz – 1 GHz Pentium III or Quad Xeon
- 500 MHz – 1 GHz Dual or Single Athlon
- Beowulf Cluster
  Distributed Memory Parallel Processing
- Myrinet Gigabit Ethernet or Dolphin Wulfkit
  High Speed, Low Latency Backplanes
- RAID



Microway Screamer™
Dual Alpha UP2000+
833 MHz, 8MB Cache in RuggedRack™
Chassis with Redundant Power Supply



Microway Scalable, 18 Node
36 Processor, Dual Pentium III
Beowulf Cluster with Myrinet and MPI

Satisfied customers have purchased Microway products since 1982. Our cost effective Linux solutions have been chosen by universities, government research labs and ISPs worldwide since 1996. Over the years Microway brand systems have become famous for their quality and price/performance. Simultaneously, our Technical Support staff earned a reputation as one of the best in the industry. This means you can count on our experience to configure a UNIX or Linux workstation or Linux Beowulf cluster solution that is ideally suited to your specific problem. This is why Microway was chosen by Los Alamos National Laboratory to maintain and upgrade their 144-node Alpha Avalon cluster. It is also why both the University of Wisconsin and Rockefeller University chose Microway to build their 100-node dual Pentium III clusters. In addition to being an Intel Product Developer and AMD Athlon OEM, Microway is **Alpha Processor, Inc.'s Top North American Channel Partner**.

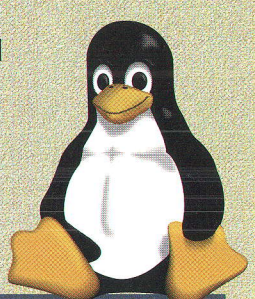
"To find a successful supplier of Linux solutions there are three things that you should know: One, the supplier must be proficient in Linux, a master of understanding Linux's superiorities; Two, the supplier must have a clear passion for hardware technology; Three, if you found a supplier that knows both Linux and understands hardware solutions, congratulations you have found Microway! When it comes to Linux and high performance, there's only one way, and that's Microway!"

— Bruce Faust, founder of DigitalScape and Carrera Computers

Microway custom configures **Linux, NT and UNIX workstations, clusters and servers** plus state of the art **RAID** systems. If you need a quality product that is fine tuned and built to last, from a company that will be around to support you for years to come, Microway is **The Number One Choice**.

Find out why over 75% of Microway's sales come from repeat customers. Please call 508-746-7341 for a technical salesperson who speaks your language!

Visit us at microway.com



Microway®

Professional Solutions for Linux Professionals

Research Park Box 79, Kingston, MA 02364
508-746-7341 • info@microway.com

Figure Two: Sorted Output of the w Command

```
[bmccarty@home bmccarty]$ w
10:06am    up 149 days, 22:01, 38 users,  load average: 0.00, 0.00, 0.00
dglyer     pts/0    faedhcp117  Thu10am  23:54m   0.73s    0.01s    /usr/local/bin/pdme
enigma     pts/22   -           7:33am   35:18    60.41s   60.41s   irc -c#tech enigma
enigma     pts/1    -           7:33am   35:40    1.16s    1.13s    irc -c#linux enigma
mccartyp   pts/21   -           9:55am   11:02    48.52s   48.51s   pine -i
philpi     pts/31   -           9:13am   5:34     0.30s    0.30s    pine
sniper86   pts/25   -           10:05am  1.00s    0.07s    0.05s    vi .te/aliases.irc
USER       TTY      FROM        LOGIN@   IDLE     JCPU     PCPU     WHAT
```

`| sort`, isn't it? Moreover, if you put the *users* file in a directory that's contained in your `PATH` environment variable, you can simply type `users`.

If this hasn't convinced you of the value of scripts, consider that scripts can contain multiple commands. Let's demonstrate by moving the pesky heading line from our last example to the top of the output, where it belongs. To do this, we simply modify our *users* script to have the contents shown in *Figure Three.*

If you execute this script, you'll see output that looks just like the original output of the *w* command, but the output appears in sorted order. In particular, the heading appears at the top of the output. How is this done? The `echo` command displays the desired heading, and the `-h` flag of the *w* command suppresses the heading that command otherwise displays.

Why stop here? Let's continue by creating a script that provides still more useful output.

Consider *Figure Four.* This supercharged descendant of the virile, yet undistinguished, *w* command displays the total number of current users as the final line of its output. This is accomplished by using the `echo` command to display an appropriate label and then using the *w* command a second time, piping its output to the *wc* command, which counts and displays the number of lines in its input. Since the *wc* command receives one line of input for each user, the result is a count of logged-in users. As you can see, Linux is truly a "Have It Your Way" operating system.

Figure Three: Fixing the Heading

```
echo "USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU   WHAT"
w -h | sort
```

Figure Four: Counting the Total Users

```
echo "USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU   WHAT"
w -h | sort
echo -n "Total current users: "
w -h | wc -l
```

Figure Five: Finding a Specific User

```
echo "USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU   WHAT"
w | grep sniper86
```

Figure Six: Finding Any User

```
echo "USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU   WHAT"
w | grep $1
```

Script Arguments

In the entertainment business, script arguments are bad news, because they can result in the loss of millions of dollars. However, in the Linux world, script arguments are good news, because they increase the flexibility of scripts.

In the context of Linux, the word *argument* doesn't refer to a dispute. Instead, it refers to a value that modifies the operation of a script, allowing you to create more flexible scripts. Suppose you are interested in running the *w* command in order to see if your friend, whose userid is *sniper86*, is logged on right now. The script in *Figure Five* would do the job.

As before, the *w* command lists the logged-on users. Here, the `grep` command filters the resulting output, discarding all lines that don't contain the text `sniper86`.

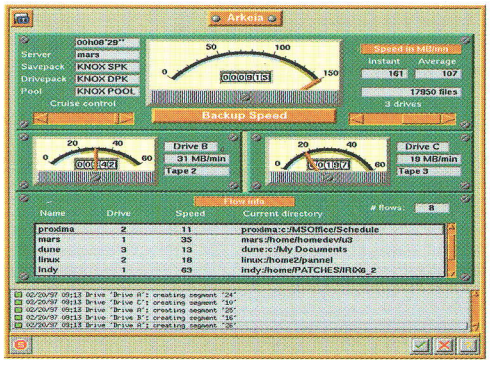
Linux Enterprise Network Backup

by Knox Software

Arkeia®. The Linux backup solution you need to keep your mission-critical data protected

Arkeia® 4.2 provides a robust and reliable Linux-based solution for your network backup needs.

- It supports over 30 major operating systems and works on over 100 different storage devices.
- It offers great speed and security. By using multi-flow technology and client side compression Arkeia® is capable of backup speeds that often exceed the network's rated speed. Its unique transaction engine allows multiple backups and restores to be performed simultaneously with total reliability.
- It features a queuing facility which allows backups to be launched when convenient rather than when resources are available.

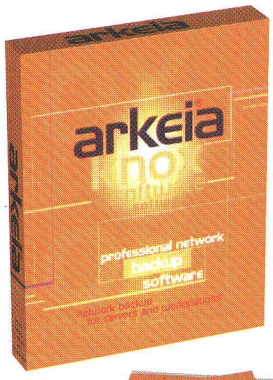


With over 30,000 users worldwide and more than 150 new companies choosing Arkeia® every month to protect their mission-critical data, Arkeia® is the de-facto standard for network backup with Linux.

Visit our web site at www.arkeia.com and benefit from a free 30 day demo of the full Arkeia® 4.2 Professional Version.

Visit our website to download a 30 day demo of Arkeia® 4.2

A free version of Arkeia® is also available for home and small business use to protect data on one Linux server and two clients.



Arkeia®...
Data protection for the real world.

www.arkeia.com

c|net Editors' Choice 2000 CNET.com

Linux WORLD 1999 Editors' Choice AWARDS

LINUX MAGAZINE Editor's Choice Award 2000

version 4.2 →

arkeia®

Figure Seven: The Contacts File

```
editors@linux-mag.com editors
linuxmag@neodata.com?subject=CustomerServiceQuery&cc=subs@linux-mag.com service
bmccarty@apu.edu Bill McCarty
bob@linux-mag.com Robert McMillan
aefrisch@lorentzian.com Aeleen Frisch
perlow@hotmail.com Jason Perlow
tech@linux-mag.com Drew Streib
merlyn@stonehenge.com Randal L. Schwartz
```

If you placed this script in the file *friend*, you could conveniently check to see if *sniper86* is logged on. But suppose you have two friends. You *could* create a second file, perhaps named *friend2*, to check whether your other friend was logged on. However, if you are sociable and have many friends, this approach quickly becomes tedious. Fortunately, there's a better way.

Consider *Figure Six* (pg. 20). Instead of containing the userid of one of your friends, this script contains the shell variable `$1`, which stands for the first argument provided by the user when the script is run. To see what this means, execute the script as follows:

```
friend sniper86
```

The command argument `sniper86` is bound to the shell variable `$1` so that the script behaves as though its second line was `w | grep sniper86` rather than `w | grep $1`.

What's cool about using an argument is that you can invoke the script in a nearly infinite variety of ways, accommodating the possibility of a practically infinite number of friends. For example, you can check whether *enigma* is logged on by issuing the command `friend enigma`. No longer is a special version of the script required for each of your friends.

To Quote or Not to Quote

When arguments can consist of multiple words, script authoring can become a tad tricky. For example, consider the data file shown in *Figure Seven*. This file, named *contacts*, contains e-mail addresses and names of *Linux Magazine* contacts.

To help you search the file, you might choose to create a script named *findcontact*, resembling the following:

```
grep $1 contacts
```

Better yet, you might include the `-i` flag so that *grep* ignores case distinctions. Also, you might use an absolute path to specify the location of the *contacts* file so that using the *findcontacts* command doesn't require you to change the current working directory to the one containing the *contacts* file:

```
grep -i $1 /root/contacts
```

To search for a contact, you might issue a command such as this:

```
findcontact bill
```

The command displays the line, or lines, in the *contacts* file that contain the text *bill*, without regard to case. In this example, it would display the line showing the e-mail address of Bill McCarty.

That's well and good, but suppose that someone with the name of Bill McMillan joins the *Linux Magazine* team, as shown in *Figure Eight*. Now the `findcontact bill` command yields two lines of output rather than a single line. Again, that's well enough. However, suppose you're interested only in the contact information for the newcomer, and so you issue the following command:

```
findcontact Bill McMillan
```

To our disappointment, the command will display two lines of output — the line for Bill McCarty and the line for Bill McMillan. The reason for this is that you've actually sent two arguments to the *findcontacts* command — `Bill` and `McMillan`. However, the script will only accept the first argument (`$1`). That is not exactly having it our way, is it?

To overcome the problem, you might attempt to enclose the arguments in quotes, converting them to a single argument consisting of two words:

```
findcontact "Bill McMillan"
```

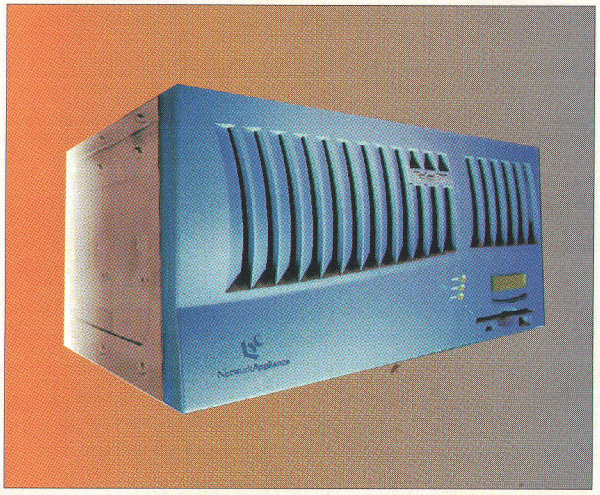
Unfortunately, this fix doesn't go quite far enough. When the shell variable `$1` in the line

```
grep -i $1 /root/contacts
```

Figure Eight: The Revised Contacts File

```
editors@linux-mag.com editors
linuxmag@neodata.com?subject=CustomerServiceQuery&cc=subs@linux-mag.com service
bmccarty@apu.edu Bill McCarty
bob@linux-mag.com Robert McMillan
aefrisch@lorentzian.com Aeleen Frisch
perlow@hotmail.com Jason Perlow
tech@linux-mag.com Drew Streib
merlyn@stonehenge.com Randal L. Schwartz
nobody@linux-mag.com Bill McMillan
```

When NETWORK APPLIANCE Users Switch To RAIDZONE OpenNAS, How Do They Spend Their Savings ... Cars? Yachts? Trips?



RAIDZONE OpenNAS 1TB	**$22,650**
Mercedes SLK320	**$40,100**
Vancouver 32 Yacht	**$62,000**
QE2 104-day World Cruise	**$57,600**
	$182,350

Network Appliance F760

1 TB

$205,945

Say you need a Terabyte of fast, reliable network storage for Windows or NFS. You could buy the NetApp 760 and be quite satisfied. Or, you could buy the RAIDZONE OpenNAS RS2000L, the Mercedes SLK320, the 32-foot sailboat, and the 104-day round-the-world cruise (in a deluxe outside cabin), and still have $23,000 left over. Which would be more than enough to buy a second TB of reliable RAIDZONE storage for your travel videos.

Come on people, this isn't rocket science. When you choose a RAIDZONE storage solution, you join our customer list alongside Boeing, Lockheed Martin, Motorola, Conexant, Palm Inc., Cisco Systems, The Dana-Farber Cancer Institute, Baylor College of Medicine, Komatsu Mining Systems, and hundreds of others.

RAIDZONE OpenNAS RS2000L File Server
$22,650

- 15 hot-swappable, 80 GB Ultra ATA/100 drives
- RAIDZONE disk array, with hot spare, hot swap, RAID 5 and on-the-fly rebuild
- Dual high performance 933 MHz Pentium III CPUs, 256MB RAM
- Gigabit & 100MBit NICs installed
- Complete browser-based remote administration interface
- Supports NFS, Windows (SMB/CIFS) and Apple clients
- Includes full 1 year warranty, with 3 years on the disks

Every one of these industry leaders has purchased a RAIDZONE OpenNAS storage solution. A few of them have already installed over 10 Terabytes of RAIDZONE storage, with more to come. Not surprisingly, none of them has had any problem deciding what to do with the large amounts of money they've saved.

A full line of RAIDZONE OpenNAS file servers is available. If your storage needs are modest, you can choose the 200 GB, 5-disk desktop cube. At the high end, we have a 45-disk, 3 Terabyte rackmount server with dual Gigabit Ethernet NICs. And every size in between. All designed so that you can get the amount of reliable, redundant storage you'll need tomorrow at a price you can afford today.

For more information, or to order online, please visit our Web site:

www.raidzone.com

or give us a call at:

1-800-388-1896

is replaced by `"Bill McMillan"`, the script operates as though the line had been written as

```
grep -i Bill McCarty /root/contacts
```

because the shell drops the surrounding quotes when it substitutes the value of the `$1` variable for its name. The result is that the *grep* searches the files *McCarty* and */root/contacts* for the text `Bill`. Of course, the file `McCarty` probably does not exist, so the output consists of both lines from */root/contacts* containing the text `Bill`. This is not what's wanted.

Fortunately, the fix is simple; both the command-line argument and the script itself must be properly quoted. Write the script like this:

```
grep "$1" /root/contacts
```

And, here's how to execute it:

```
findcontact "Bill McMillan"
nobody@linux-mag.com Bill McMillan
```

Although it required some ingenuity to get here, the script now behaves as desired. Linux's reputation as a have-it-your-way operating system is secure.

The Long and Winding Road

This column, as usual, has scarcely touched the surface of its topic, shell scripts. Among the most useful shell features not covered are multiple arguments, more sophisticated rules and methods of quoting arguments, environment variables, conditional execution, looping, backquoted expressions, and signals. Future *Newbies* columns will touch on these stepping-stones to Linux mastery.

The topical development of this column was inspired by Chapter Three of my favorite Unix book, Brian W. Kernighan's and Rob Pike's *The Unix Programming Environment* (Prentice-Hall, 1984). If you're rushing down the road to Linux gurudom, you need to get and read this book. It's a terse, and therefore somewhat tough, read that demands concentrated attention, but your efforts will be rewarded many times over. Until next month, happy scripting!

Bill McCarty is an associate professor at Azusa Pacific University. He can be reached at bmccarty@apu.edu.

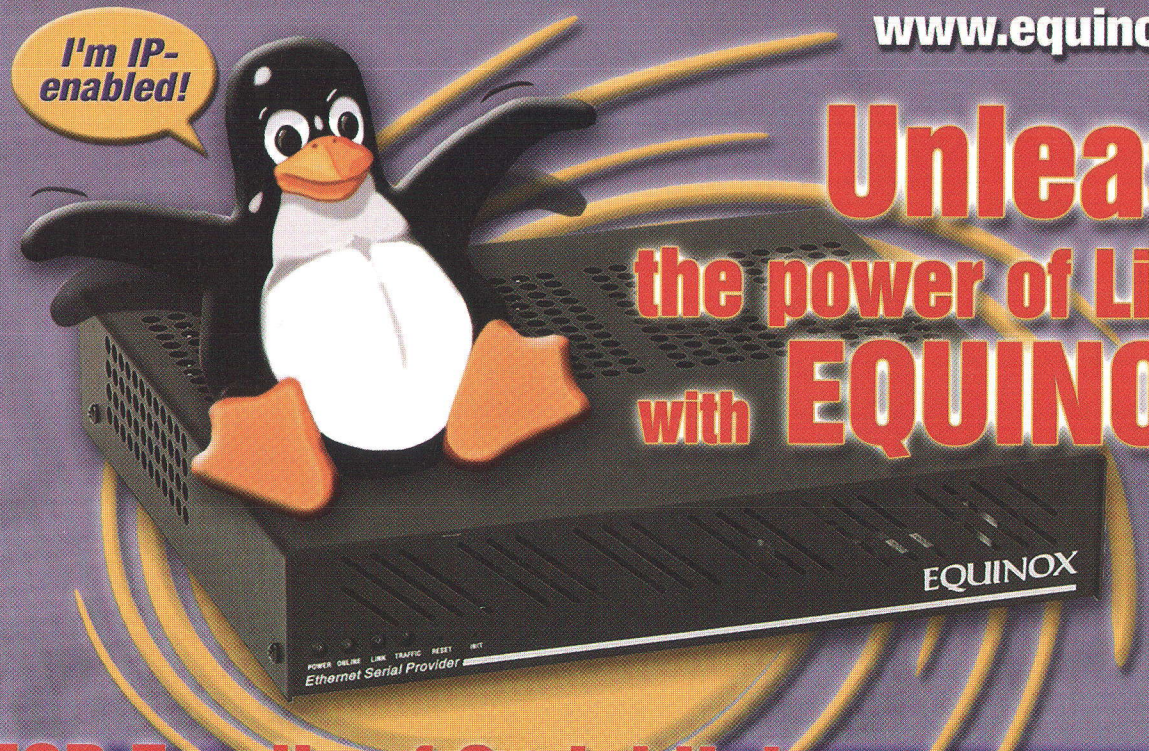
Finding a Good Script

As everyone knows, the real-life actors we mentioned at the beginning of this article often have trouble finding a good script. In fact, some actors have been known to resort to writing scripts especially designed to showcase their talents. Okay, so maybe we're stretching the actor analogy a bit thin here, but the point is that you too can write your own scripts. However, to be able to write effective scripts, you'll need to employ a wide-ranging repertoire of Linux commands. The table below summarizes some Linux commands that are often used in scripts. Consult the man pages or other documentation to learn more about these commands.

Important Commands Often Used in Scripts

at	Execute a command at a specified time
cat	Display contents of a file
chmod	Change file permissions
comm	Compare files
cp	Copy files
df	Display disk space usage of filesystems
diff	Compare text files
du	Displays disk space usage of directories
echo	Display text
eval	Evaluate an expression
grep	Find strings
kill	Terminate a process
lpr	Print data
less	Page through a file
ls	List file information
mail	Send a mail message
mkdir	Create a directory
mv	Move a file or directory
nice	Set process priority
nohup	Run a command immune to hang-ups
od	Display file contents in octal and other formats
passwd	Set a user's password
pr	Format data for printing
ps	Display process information
rm	Remove a file
rmdir	Remove a directory
sh	Launch a shell
sleep	Pause a process
sort	Sort data
spell	Check spelling
tail	Display last lines of file
time	Time the execution of a script
times	Displays CPU time used
uniq	Eliminates duplicate data
w	Displays logged in users
wait	Waits for child processes to terminate
wc	Counts characters, words, and lines
who	Displays logged in users

www.equinox.com



Unleash the power of Linux with EQUINOX!

ESP Family of Serial Hubs

With the ESP family of 10/100 serial hubs you can attach devices anywhere on your Ethernet network, local or remote.

ESP Serial Hub
- 8 or 16 port serial hubs
- 8-wire RS-232 interface
- Full modem controls

ESP Multi-Interface
- 8-port serial hub
- Soft selectable serial interfaces;
- RS-232 • RS-422 • RS-485

- **Retail POS**
- **Security**
- **Dial Access**
- **Process Control**
- **Data Collection**
- **Fax Server**

Ethernet serial connectivity solutions from EQUINOX

Network-based

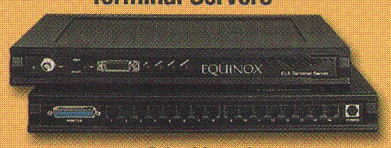
Managed Device Server
MDS-10

Single port device server

ESP Serial Hubs

Available in 8 or 16 serial ports

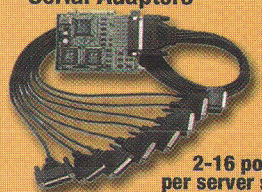
Terminal Servers



8 to 16 ports

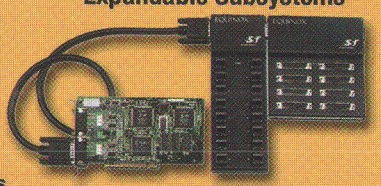
Bus-based

Multiport Serial Adapters



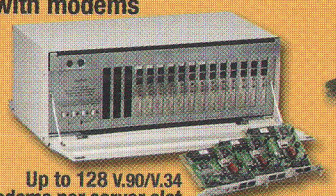
2-16 ports per server slot

Expandable Subsystems



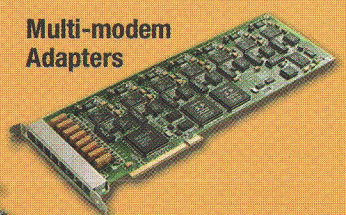
Up to 128 serial ports per server slot

Analog Modem Pool with modems



Up to 128 V.90/V.34 modems per server slot

Multi-modem Adapters



4 or 8 modem adapters V.90/56K

For a FREE 30-day evaluation of any Equinox product, call 800-275-3500 ext. 607 or email sales@equinox.com

EQUINOX®
an Avocent Company

IN THE TRENCHES

Linux: The Dawn of a New Day

By Ransom Love

With the recent downturn in the economy and the drop in Linux companies' share prices, an uninformed observer might believe that the sun is setting on Linux. They would be wrong. Unfortunately, the expectations that had previously been set for Linux are analogous to parents expecting that their teenager successfully complete college before graduating from high school. Few, if any, possess the maturity to make that jump.

The failures teenagers encounter as they strive to cross the chasm between adolescence and adulthood do not diminish their potential in our eyes. In fact, we applaud and encourage them to continue to try again, because we know that those who are the brightest and most capable will have more failures before they reach their potential.

As a father of seven children (three of whom are teenagers) and a founder of Caldera, I know whereof I speak. The Linux industry is as complex and difficult to understand as a teenager. The challenges facing Linux have nothing to do with its potential and everything to do with its maturity. With that in mind, I'd like to make five suggestions for the Linux industry:

- Learn to honestly evaluate successes and failures
- Uphold standards
- Develop and/or embrace mature business channels outside of retail
- Specialize
- Seek compatible companions that will help Linux mature faster

Evaluate Successes and Failures

The Linux industry and the Open Source development model have had tremendous successes that have changed the information technology landscape for good. The Open Source software development model facilitates collaboration and innovation in an unparalleled manner. Linux, as a technology, has matured much faster in its development than any other technology or operating system. However, Open Source is not without its problems — even as a development model.

An Open Source model does not provide a complete road map of where the technology is going. The outcome and timing of Open Source development is much less predictable. Open Source is akin to the ideal sandbox for a developer — lots of technology to play with and no one giving assignments or deadlines. This is great for development but not great for business.

Open Source has made a significant contribution to the way software is and will be developed. It has proven to be a method of development worthy of further review and refinement. We can mature the model by developing road maps and improving timelines.

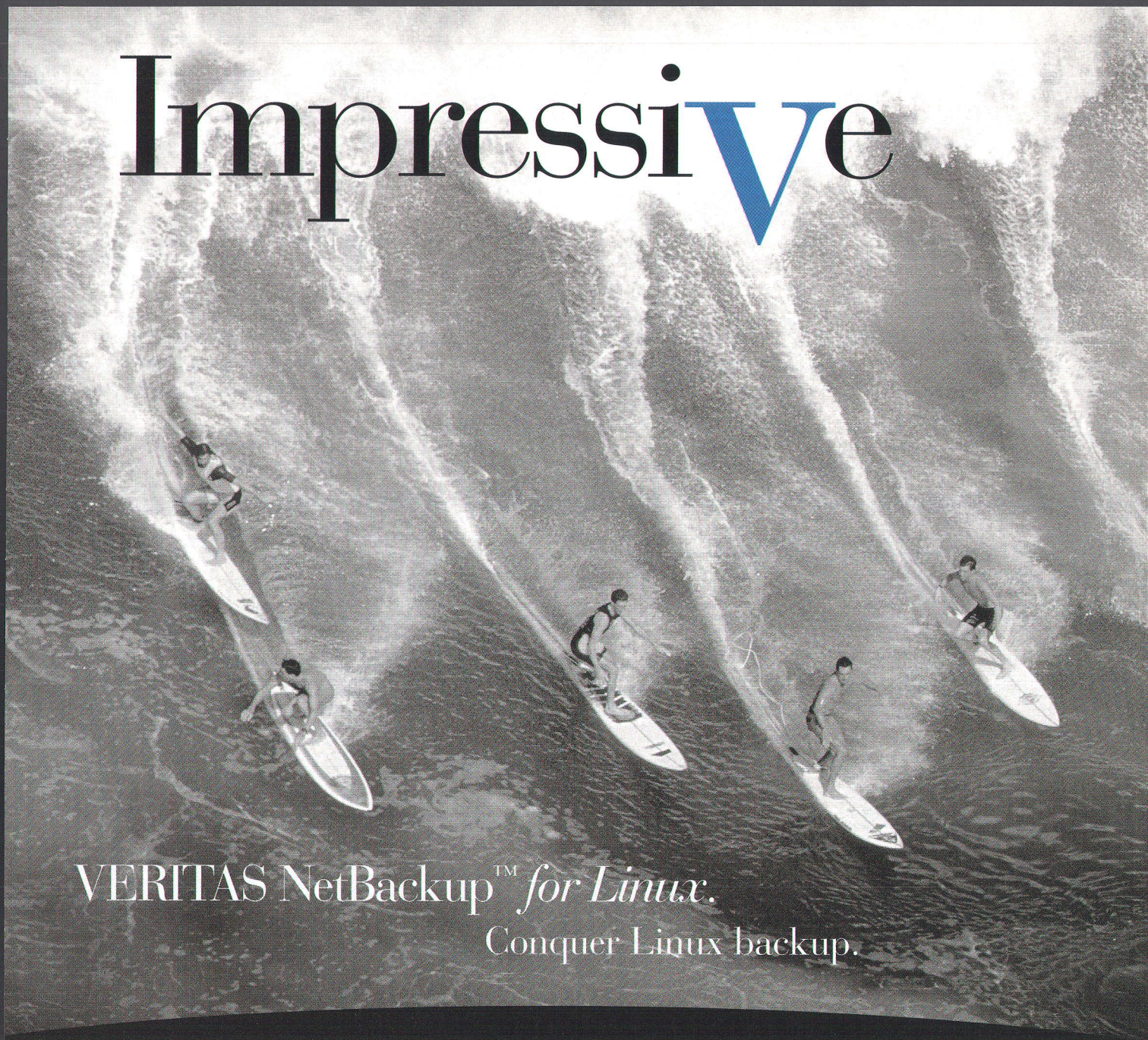
While the Open Source development model shows great potential, the business model needs some additional work. If we want to support the development model of "free" software, we must come to realize that someone has to make money. If the Linux providers cannot become profitable, the only companies who will continue to invest in Linux will be those with a differing agenda. Their income will be derived from sources other than Linux itself, and their primary allegiance will be to those other sources of revenue.

Linux providers must be free to experiment with business models that allow them to earn a profit. The Linux market must mature into a real industry or it will be relegated to acting as a catalyst for existing industries like hardware or software. The community must come to terms with the fact that the Linux providers need to make money in order to stay healthy. Without healthy Linux providers, we run the risk of returning to the bad old days of fragmented Unix, and we potentially allow the underlying hardware architecture to drive the platform and differentiation.

TONY KLASSEN

Uphold Standards

So, what is needed for Linux to mature as an alternative business platform? Well, continuing with our maturing teenager example — young adults eventually come to realize that



VERITAS NetBackup *for Linux*. The Value of Leadership.

The #1 provider of backup solutions in America is the natural choice for your Linux environment. With NetBackup *for Linux*, VERITAS offers a true enterprise-level backup and recovery solution. It's advanced, yet intuitive and easy to use; comprehensive, yet remarkably affordable. And because NetBackup supports all major operating systems, hardware platforms and applications, it's the premier backup solution for a heterogeneous world. Then consider the proven performance, scalability and VERITAS quality support, and it's clear that NetBackup *for Linux* delivers value you thought was beyond your reach.

For more information on VERITAS NetBackup *for Linux*, visit www.veritas.com/netbackupfamily/linux.

VERITAS™

BUSINESS WITHOUT INTERRUPTION™

1-800-729-7894 code: 8730110AI | veritas.com

laws or standards are for their benefit, not to simply create hindrances or restrictions. In much the same manner, Linux needs to have a standard reference platform and distribution that all developers can write to.

The industry has talked about adopting a Linux Standard Base (LSB), but the early success of Linux has created obstacles to its adoption. Every Linux provider felt that they had an opportunity to become the dominant platform, and the tight development cycles that are required to remain competitive in the retail space, combined with the average developer's insatiable desire to have the latest and greatest technology, fueled a self-feeding retail frenzy.

Develop and Embrace Mature Business Channels

This leads to the next point. Just as a teenager must branch out beyond his parents in order to integrate the knowledge and experience of others, Linux must branch out from its retail roots and develop mature business channels. Mature business channels like Value Added Resellers (VARs), Consultants, and Original Equipment Manufacturers (OEMs) are critical to the business customer. These specialists provide total solutions for their clients. No one buys an operating system — they buy a solution to their business needs.

Because we live in a global economy, these channels and the companies that support them must be global. Having regional Linux companies is a major headache for business customers as well as for major application and hardware vendors. While many large application and hardware providers support multiple regional Linux providers, most smaller businesses cannot.

Consequently, a Linux provider must be able to offer a global infrastructure for support, marketing, and sales. To build a global infrastructure requires tremendous resources, time, and capital. With money tight, the current market is forcing the maturation and consolidation of suppliers.

Specialize

Just as a teenager must seek specialized training, Linux providers must begin to focus and specialize. Rather than view each other as competition, we can partner in key areas that would be beneficial to the industry at large. Since there is little or no money in a base distribution of Linux, we should all provide an LSB-compliant distribution. Specialization can then come from the markets we choose to support or the product areas above the operating system in which we develop intellectual property.

Seek Compatible Companions

Finally, when we are young, we feel that we do not need anyone's help to achieve our goals. We have no fear. When we get a little older and wiser, we realize that partnering with others can provide tremendous growth, comfort, and enjoyment. There are those in the Linux community who believe that they can do it all themselves. However, they are unrealistic.

The Linux industry and the Open Source development model have had tremendous successes that have changed the IT landscape.

With all of the investment capital that has been poured into it, and with all of the media interest surrounding it, there is tremendous pressure on Linux (as a technology and a business model) to mature much faster than may be realistic.

At Caldera, we believe that unifying SCO Unix with Linux is a way of achieving that maturity in an accelerated time frame. SCO Unix as a technology has test suites, application programming interfaces, tools, and infrastructure that would be very compatible with, and complementary to, Linux. This unified system will enable application developers and business customers to more fully embrace and deploy Linux into the mainstream business back office. There is clearly a trend in the industry of traditional Unix providers integrating Linux with their current offerings. The Linux industry should embrace these groups.

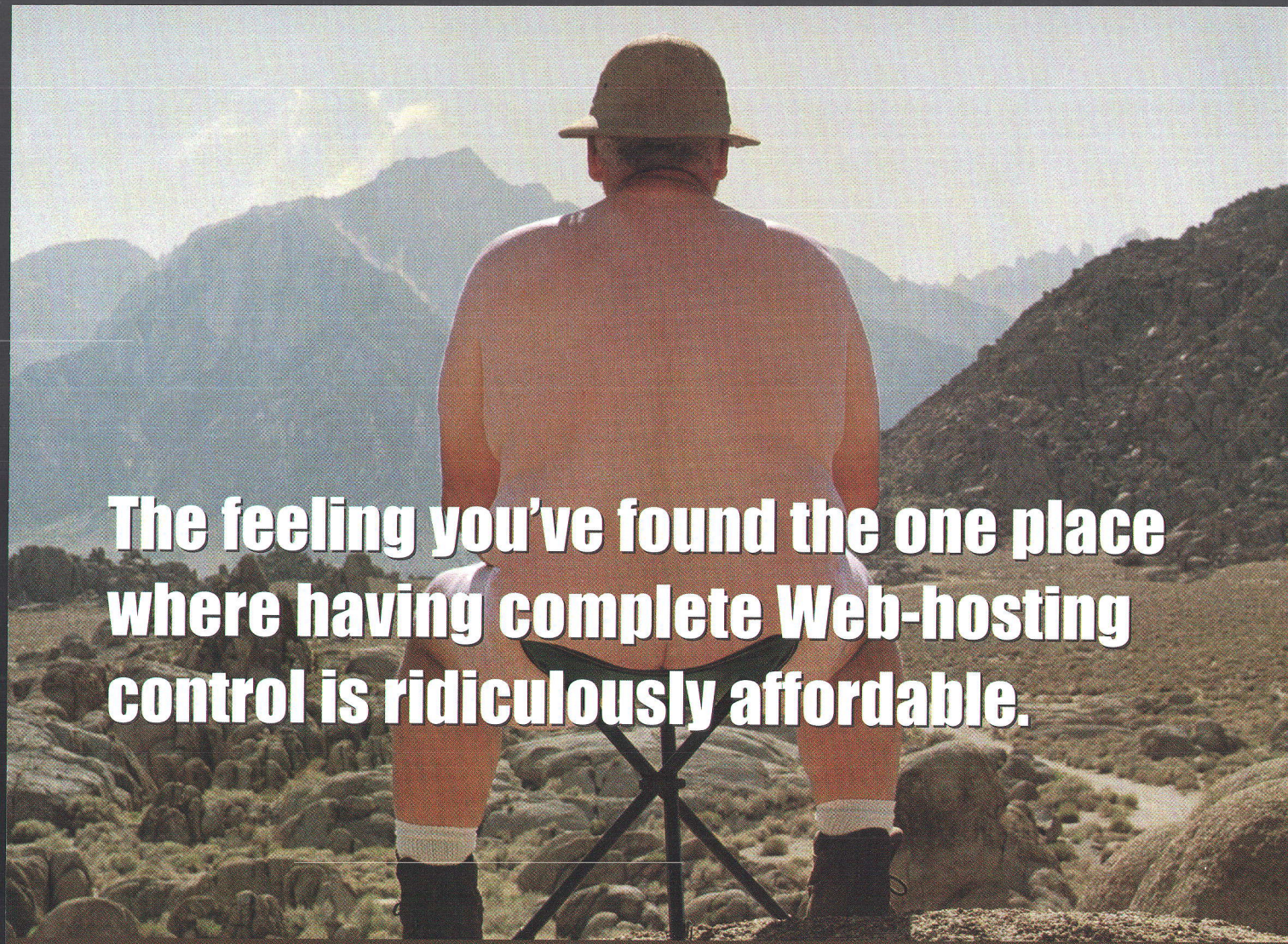
There are things in the Unix industry that need to change, and there are things in the Linux industry that need to change. However, their marriage could result in the most reliable, scalable, open, and useful technology platform the industry has ever known.

My greatest joy as a father has come from watching my teenagers, in both their failures and triumphs, take on the additional challenges of adulthood. For those who conclude from my analogy that they should wait to embrace Linux until it matures, I issue this challenge. If you wait, you may be left behind. Linux has grown and matured faster than any other operating system platform in history. Many of the issues that I raise in this piece are being addressed even as I write these words.

Like youth itself, Linux is fresh. The development model is new and intriguing. The business model is not yet fully formed, but this provides opportunity. Linux is a "disruptive" technology; it has and will continue to change the information technology world. More importantly, like the Internet itself, Linux will change the way we do business, the way we develop and deploy technology, and the way we package and deliver applications and business solutions.

Get to know this teenager called Linux and you will be richer and wiser, financially and personally. Get to know Linux and see the potential for change in our industry. Linux is the smart alternative, the dawn of a new day.

Ransom Love is CEO of Caldera Linux Systems, a publicly-held Linux distribution and services company.



Introducing FREEDOM. More power for less, for once.

It's called HostPro Freedom for a reason. For the first time ever, HostPro gives you more control than servers costing three times as much. We're the only host to offer you four unique features for Unix platforms. First, you have complete control over applications. HostPro Freedom lets you run just about any software you want, or even change the Web server. Second, customization is no problem, since you're running open source FreeBSD and have access to a port library of 4000 tested apps. Unlimited shell accounts allow you to create and control multiple users. Third, you get multiple domain hosting for just $149.95 a month, with up to 40 domains hosted at no additional monthly charge. And finally, HostPro Freedom's Virtual Integrity technology adds unique stability to your site. Each account acts as though it has its own hard drive – a virtual disk, as we like to call it. That means your shared server stays stable even when other users run their own applications.

Impressive? We think so. No one else offers all this freedom. Call us for more information, or go to HostPro.com. And then do anything you want.

Specifications	FREEDOM 200	FREEDOM 300	FREEDOM 400x	FREEDOM 500x
Price	$59.95	$99.95	$149.95	$249.95
Disk Space	300MB	400MB	500MB	1GB
Bandwidth	10GB	20GB	30GB	40GB
Email Accounts	*Unlimited	*Unlimited	*Unlimited	*Unlimited
Shell Access	Yes	Yes	Yes	Yes
Root Access	No	No	Yes	Yes
FTP/Shell Users	1	10	*Unlimited	*Unlimited
Multiple Domain Hosting	10 Domains	20 Domains	40 Domains	*Unlimited
Shopping Cart	Miva Merchant	Miva Merchant	Miva Merchant	Miva Merchant
SSL Certificate	Yes	Yes	Yes	Yes

(visit our website for more) *Subject to physical hardware limitations

HOSTPRO®

Try us on for size.

1-800-862-8917

hostpro.com – enter code: BIGGUY4

BIGGUY4@hostpro.com

Understanding the NETWORK Neighborhood

How Linux Works With Microsoft Networking Protocols

by Christopher R. Hertel

SMB, the Server Message Block Protocol, is the most prevalent filesharing protocol on the planet for a very simple reason — it ships with every Microsoft Windows system and, like it or not, Windows still owns the desktop. Windows is also very common as a server platform in corporate networks. Not content with those markets, Windows is now finding its way into all sorts of new places, including embedded systems, palmtops, and consumer toys. As Windows moves onto new platforms, SMB does too.

Open source operating systems like Linux have been speaking SMB for quite a while now thanks to SAMBA, the well-known open source SMB Server suite. SAMBA, like Windows server products, is primarily a back-room tool. It runs on systems that are mounted in racks or stuck onto shelves in locked server rooms where only the geeks are brave enough to go. If Linux is going to move out of the datacenter and onto the corporate desktop (not to mention homes, hand-helds, cars, etc.), then Linux developers are going to need a working knowledge of SMB — the native language of the Microsoft Network Neighborhood.

In this article, we'll look into SMB's history and architecture, as well as how its components work together. You'll also find a list of open source projects that aim to make it easier to add SMB support to Linux applications.

A Little History: NetBIOS

SMB was originally intended to run over a proprietary network system co-developed by IBM and a company called Sytec. In a moment of obvious inspiration, this system was dubbed "PC-Network." It had no support for routing and could only handle a maximum of about 80 nodes. It was truly LAN-locked.

PC-Network was a broadband LAN product consisting of network cards, cables, and a small device driver known as NetBIOS (Network Basic Input/Output System). The original PC-Network hardware is long gone, having been replaced by Token Ring and then Ethernet. Unfortunately, lots and lots of software was written for use with the NetBIOS Application Programmer's Interface (API), so, even though the PC Network hardware is no longer in use and the NetBIOS device driver is no longer needed, the NetBIOS API has remained as a living artifact.

Instead of moving away from NetBIOS and letting it die an honorable death, several vendors implemented the NetBIOS API on top of other protocols, including DECnet, IPX/SPX, SNA, and TCP/IP. NetBIOS over TCP/IP is often called NBT and has become the preferred NetBIOS transport. The workings of NBT are described in two Internet Engineering Task Force (IETF) Request For Comment (RFC) documents, RFC1001 and RFC1002 (known collectively as Internet Standard #19). NBT is pictured in *Figure One.*

SMB/CIFS

The SMB protocol was designed to run on a PC-Network LAN, using the NetBIOS API to send and receive packets. This did not change until the release of Windows 2000 (W2K), the first version of Windows to support SMB packet transport over TCP/IP without NetBIOS encapsulation. Even so, W2K includes NBT support to maintain compatibility with its predecessors. SMB over TCP/IP is shown in *Figure Two.*

SMB was originally developed by Intel and Microsoft in the early 1980s and has been the core of DOS and Windows filesharing ever since. Some time around 1996, as part of the buildup to W2K, Microsoft executed a Marketing Upgrade on SMB and renamed it CIFS, or Common Internet Filesystem.

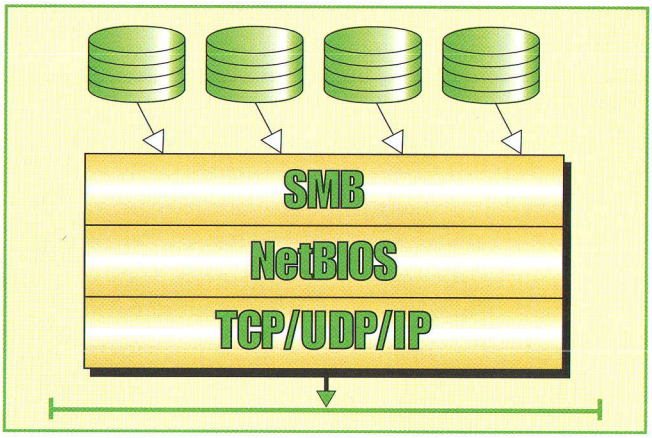


Figure One: NetBios running over TCIP/IP.

CIFS enables the sharing of directories, files, printers, and other cool computer stuff across a network. To make use of these shared resources you need to be able to find and identify them; you also need to control access so that unauthorized users can't fiddle about where they aren't allowed. This means there is a hefty amount of administration to be managed, so CIFS filesharing comes with an entourage. There are protocols for service announcement, naming, authentication, and authorization. These are separate but intertwined. Some are based on published standards, others are not; most have changed over the years. These days, the term "CIFS" is most often used to refer to the full suite, while "SMB" is typically used when discussing the filesharing protocol itself.

In 1997, Microsoft submitted draft CIFS specifications to the Internet Engineering Task Force (IETF). Those drafts have since expired, but there is an effort underway by the Storage Network Industry Association (SNIA) to revive and overhaul them outside of the IETF process.

How It Works

Because of its heritage, the CIFS suite can be a bit awkward. Most of the silliness exists at the NetBIOS layer because, as we have already explained, NetBIOS is an anachronism.

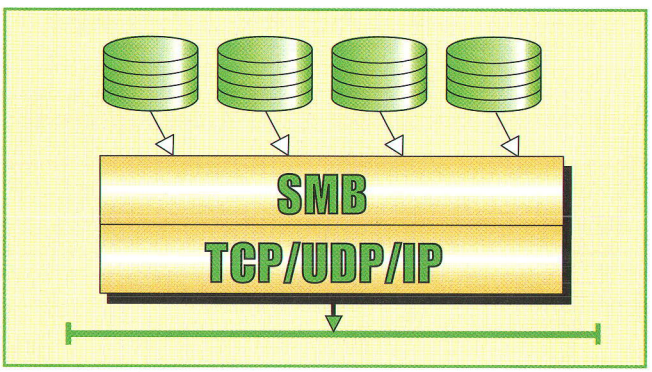


Figure Two: SMB running over TCIP/IP.

NBT is an implementation of the NetBIOS API on top of TCP/IP, but what RFC 1001 & 1002 actually describe is a system for emulating NetBIOS-based PC-Network LANs over a routed IP inter-network. This is critical to understanding the workings of NBT — it is a virtual LAN system. The nodes in a CIFS filesharing network are connected to an imaginary wire by imaginary network adapters. It's all make-believe.

There are three key parts to NBT. These are:

- the Name Service
- the Datagram Distribution Service
- the Session Service

The Name Service handles NetBIOS names (the addresses used on the emulated PC-Network LAN). The Datagram Distribution and Session Services carry data between the nodes on the virtual PC-Network LAN.

The NetBIOS Name Service

Each NetBIOS name is a communications end-point, representing an application or daemon that is waiting to hear from other applications or daemons across the virtual wire. The Name Service keeps track of which names are in use at which IP addresses, thus allowing the underlying IP network to find the nodes and transport NetBIOS messages between them. The Name Service runs on UDP port 137.

There are two kinds of names — group and unique. Group names can be shared so that datagrams can be multicast to all members of the group. In contrast, only one instance of a unique name may be registered at a time within a virtual PC-Network LAN.

The Name Service has two modes of operation — broadcast and point-to-point. In broadcast mode, names are registered, queried, and eventually released by sending UDP broadcasts to port 137. It's sort of like calling out, "Yo! Anybody here named RUGRAT?" in a crowded room. If there is a RUGRAT in the room, you would expect an answer like, "Yeah, here I am."

Point-to-point mode is used to cross IP subnet boundaries. Since IP broadcasts are typically limited to local IP subnets, a special server called the NetBIOS Name Server (NBNS) must be used to coordinate and manage the names in use on a given NBT virtual LAN. All registrations, queries, and releases are sent directly to the NBNS, which keeps the name-to-IP mappings in a database. Microsoft's NBNS implementation is called WINS (Windows Internet Name Service). The term WINS is now commonly used instead of NBNS, but we will be pedantic and stick with the latter.

It is possible, and even common, to combine broadcast and point-to-point name management. The RFCs describe "Mixed mode," and Microsoft later added "Hybrid mode." These two modes differ only in the order in which the broadcast and point-to-point mechanisms are applied.

The NetBIOS Datagram and Session Services

Data transport is handled either by the Datagram Distribution Service or the Session Service, depending upon the needs and design of the application. In the IP world, TCP provides connection-oriented sessions in which packets are acknowledged, put in order, and retransmitted if lost. This creates the illusion of a continuous, sequential data stream from one end to the other. In contrast, UDP datagrams are simply sent. UDP requires less overhead, but it is also considerably less reliable than TCP.

NetBIOS also provides connection-oriented (session) and connectionless (datagram) communications. Naturally, NBT

Alphabet Soup

The following is a list of acronyms relevant to the Microsoft Networking Protocols:

CIFS	Common Internet File System
DMB	Domain Master Browser
DNS	Domain Naming Service
IPX/SPX	Internetwork Packet Exchange/ Sequenced Packet Exchange
LMB	Local Master Browser
NBDD	NetBIOS Datagram Distribution Server
NBNS	NetBIOS Name Server
NBT	NetBIOS over TCP/IP
NetBIOS	Network Basic Input/Output System
SMB	Server Message Block Protocol
SNA	Systems Network Architecture
TCP/IP	Transmission Control Protocol/ Internet Protocol
UDP	User Datagram Protocol
W2K	Windows 2000
WINS	Windows Internet Name Service

uses TCP to carry NetBIOS sessions and UDP to carry NetBIOS datagrams. These services run on 139/TCP and 138/UDP, respectively.

The Datagram Service

Sending a datagram to a unique name is fairly simple. The name is resolved to an IP address via the Name Service, and the NetBIOS message is tucked into a UDP packet that is sent to port 138. That's it.

Sending a multicast (group name) datagram is also fairly simple if broadcast mode name management is in use. In this case, group datagrams can be sent to the IP broadcast address instead of a unicast address. All local nodes will see the packet, but only group members will actually open it. It's not too tough.

If the NBT virtual LAN crosses IP subnet boundaries, however, sending NetBIOS datagrams to a group name gets a bit icky. Per the RFCs, the same system that is running the NBNS also runs a service called the NetBIOS Datagram Distribution Server (NBDD); multicast datagrams are sent to the NBDD, which gets the list of IPs associated with the group name from the NBNS; the NBDD then forwards the datagram individually to each group member. It's sort of like sending a group e-mail to a mailing list server. You send one message, and the server takes care of distributing copies to all of the recipients.

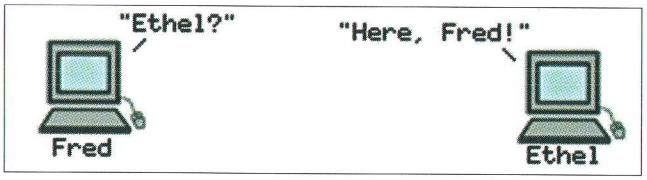
The problem with the datagram service is that Microsoft messed it up. They made a mistake when they implemented WINS. With the exception of one special case, WINS fails to keep track of IPs associated with a group name. Instead, WINS stores only the generic broadcast address 255.255.255.255. Because of this, Microsoft never bothered to implement the NBDD. The upshot is that some group members will not receive group multicasts, which has implications for services that rely on group names. We will see an example of this later on when we examine the Browser Service.

The sad truth is that SAMBA, in an effort to remain compatible, followed Microsoft's example.

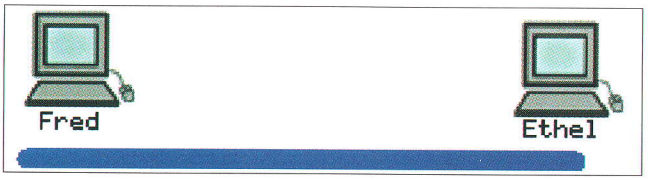
The NetBIOS Session Service

Under NBT, NetBIOS sessions are created on top of TCP sessions. Here's what happens when node FRED tries to establish a NetBIOS session with node ETHEL:

- FRED uses the Name Service to find the IP address of node ETHEL.

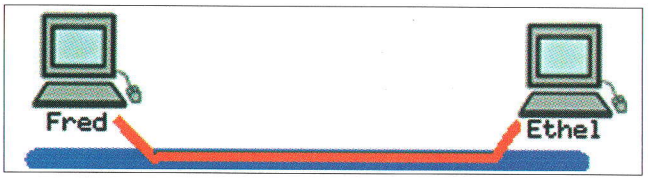


- FRED establishes a TCP connection to TCP port 139 on node ETHEL.



- FRED sends an NBT SESSION SERVICE REQUEST packet via the TCP connection. The request contains the NetBIOS name of the source node (FRED) and the NetBIOS name of the target node (ETHEL).

- The SESSION SERVICE REQUEST can be rejected if ETHEL isn't home (that is, the software that registered ETHEL is not actually listening or the name was never registered at all). If the request is accepted, the two systems may send NetBIOS session packets via the TCP tunnel until the connection is closed.



The Session Service is the simplest of the three NBT services. It does not need to worry about distributing messages to all owners of a group name since it is inherently a point-to-point service. It is, however, the transport for SMB file-sharing, so it is of particular interest to us.

The Sum of the Parts

The purpose of NBT is to provide an emulated PC-Network LAN. It does not matter if the participating nodes are scattered across the Internet. If they share the same NetBIOS name space, they are on the same virtual wire. It is the Name Service that is responsible for creating and maintaining the name space, so the Name Service defines the virtual LAN.

The NetBIOS API is the gateway to that virtual LAN, but non-Windows systems generally avoid using that interface. Instead, they typically craft the NBT packets and handle TCP and UDP transfers directly. This, unfortunately, can give the impression that SMB and its associated services are all IP-based, which really isn't the case. Remember that the NetBIOS API has been implemented on top of lots of other protocols too.

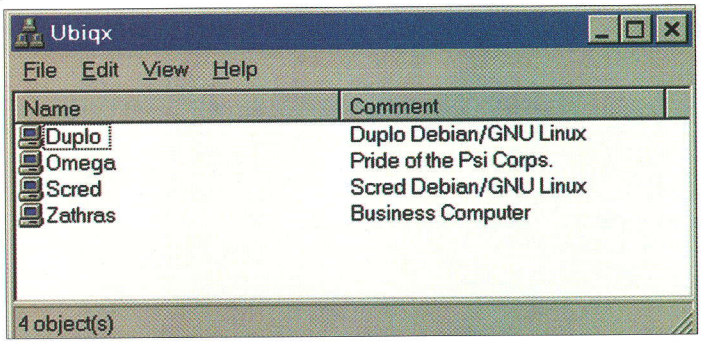


Figure Three: Servers available on the network.

SMB

Now that we have jumped through the flaming NBT hoops, it's time to juggle the SMB chainsaw.

The first thing to note about Server Message Block is that SMB packets use Intel little-endian byte order, while NBT uses big-endian network byte order. No matter how you fiddle with it, if you are going to implement SMB, you are going to have to swap a few bytes.

The Server Message Block is a record structure. The first field always contains the identifying characters '0xFF' 'S' 'M' 'B', just to make it absolutely clear what you are dealing with. The second field is the command. SMB messages are made up of a command, the data associated with the command, and the context in which the command is to execute. The context information allows SMB to keep track of multiple links multiplexed within a single NetBIOS session.

Most of the SMB commands are derived directly from DOS I/O functions. They include basic stuff like OPEN, CLOSE, and DELETE, plus commands for handling print jobs and a few other oddities. Before these can be used, however, a client must gain access to a shared printer or directory (share). However, the SMB protocol has undergone a bit of evolution since it was first introduced, and this has resulted in a number of "dialects." To accommodate the various SMB dialects, there is a NEGOTIATE SMB that lets nodes discuss and agree upon an SMB dialect to use.

Documentation on SMB can be found on Microsoft's FTP servers. Just dig around in ftp://ftp.microsoft.com/developr/drg/cifs/ for a while if you are curious. Start with the older stuff and work your way forward.

For Further Reading

For additional information about SAMBA and Windows networking protocols, check out these two articles from previous issues:

- http://www.linux-mag.com/1999-05/samba_01.html
- http://www.linux-mag.com/1999-09/samba_01.html

Presentation Is Everything

Now it's time to put a pretty face on all of this. On any Windows desktop, you will likely find an icon labeled "Network Neighborhood." This is the front door to the CIFS (and SMB) world. Double-click the icon and you should see something resembling *Figure Three.*

The icons in the window represent servers available on the network. (Your network, of course, will have different servers listed.) Double-click a server icon and you should see a list of the shares offered by that server. *Figure Four* contains such a list.

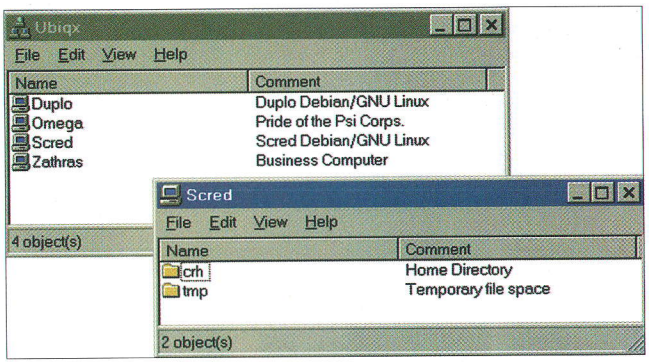


Figure Four: File shares available on the Scred server.

Sure Looks Pretty, Doesn't it?

The underlying system that makes this presentation possible is called the "Browser Service." This service collects and maintains the "Browse List," and viewing the Browse List (e.g., via the Network Neighborhood) is called "Browsing." It should be noted that Microsoft came up with these names before the invention of the Web Browser, so they cannot be blamed for any ensuing confusion.

Browsing is organized in terms of IP subnets and Workgroups. A "Workgroup" is a set of NBT nodes on an IP subnet that shares the same Workgroup name. In our examples, all of the nodes are members of the UBIQX workgroup.

On each subnet, the Workgroup members hold an "election," which involves sending group datagrams via the NBT Datagram Service. The election mechanism makes Florida recounts look easy, so we will save the description for another day. Eventually, a winner is declared and designated as the local "Master Browser" (LMB) for the Workgroup. If there are a lot of nodes in the Workgroup, additional local Browsers may be elected to serve as "Backup Browsers."

When a client wishes to see the Browse List, it asks one of the Browsers on the local LAN for a copy; this is what is displayed when you double-click Network Neighborhood.

As described earlier, the lack of a working NBDD in Microsoft's implementation of NBT limited browsing to IP subnets. Microsoft recognized the need to circulate Browse Lists outside of IP subnets, so they created yet another new serv-

Dual 1GHz CPU 1U Rackserver for $1,995!

2 GHz Under 2K!

Most cost-effective Dual CPU 1U!
- Dual Pentium III CPUs, up to 1GHz
- 128MB Memory, up to 1.5GB
- 30GB IDE Hard Drive
- Intel 10/100 Ethernet, up to 4
- 1U Rack height (1.75"H), fit up to 84 CPUs/42 Servers in a Rack
- Standard with Red Hat Linux 7, also supports Windows NT/2000
- Get Cobalt-like Server Appliance feature with optional Browser Manager™ (see sidebar) for $399 more!

V-Raptor 1U Dual 1GHz
Dual Intel Pentium III 1GHz
$1,995 *$76/mo*

Dual Intel Pentium III 750 MHz
$1,595 *$59/mo*

Dual Intel Pentium III 866 MHz
$1,795 *$68/mo*

Standard Features include Dual CPU Ready Pentium III Motherboard, Intel 82559-based 10/100 Fast Ethernet Adapter[3], Optional Intel 10/100 Fast Ethernet Adapters (up to 4) @ $69/ea add'l, PS/2 Keyboard & Mouse/USB/Serial/Parallel Ports, 1U Rackmount Aluminum Chassis (2 HD, 1 CD/FD/HD Drive Bays), 200 Watt 1U PS, Einux Compact Cooling System, Red Hat Linux 7 OS, Also supports Unix, Windows NT/2000 OS. Dimensions: 1.75"H x 19"W x 23"D

Lowest cost Dual Ready 1U Rackserver!
- 1 to 2 Pentium III CPUs, up to 1GHz
- 128MB RAM, Max 1.5GB
- 20GB IDE Hard Drive
- Intel 10/100 Ethernet, up to 4
- Standard with Red Hat Linux 7, also supports Windows NT/2000
- Get Cobalt-like Server Appliance features with optional Browser Manager™ (see sidebar) for $399 more!

Phoenix 1U Dual CPU Ready
Celeon 700 MHz
$995 *$38/mo*

Pentium III 750MHz
$1,195 *$45/mo*

Pentium III 866 MHz
$1,295 *$49/mo*

Pentium III 1 GHz
$1,495 *$54/mo*

Upgrade to 256MB $99

Upgrade to 40GB HD $99

Standard Features include Dual CPU Ready Pentium III Motherboard, Intel 82559-based 10/100 Fast Ethernet Adapter, 3 Optional Intel 10/100 Fast Ethernet Adapters (up to 4) @ $69/ea add'l, PS/2 Keyboard & Mouse/USB/Serial/Parallel Ports, 1U Rackmount Aluminum Chassis (2 HD, 1 CD/FD/HD Drive Bays), 200 Watt 1U PS, Einux Compact Cooling System, Red Hat Linux 7 OS, Also supports Unix, Windows NT/2000 OS. Dimensions: 1.75"H x 19"W x 23"D

First AMD 1U & Fastest 1U Available!
- Athlon/Duron CPUs, up to 1.2GHz
- 128MB RAM, Max 1.5GB
- 2 x 30GB IDE Hard Drive (RAID 0/1)
- Hardware IDE RAID on-board
- Intel 10/100 Ethernet, up to 4
- Standard with Red Hat Linux 7, also supports Windows NT/2000
- Get Cobalt-like Server Appliance features with optional Browser Manager™ for $399 more!

T-Bird 1U AMD with RAID
AMD Duron 750MHz
$1,195 *$45/mo*

AMD Duron 850MHz
$1,295 *$48/mo*

AMD Athlon 1 GHz
$1,495 *$54/mo*

AMD Athlon 1.2 GHz
$1,695 *$59/mo*

Upgrade to 256MB $99

Upgrade to 2 x 40GB HD $199

Standard Features include AMD Duron/Athlon Motherboard with on-board Hardware RAID 0,1,0+1 IDE Controller, Intel 82559-based 10/100 Fast Ethernet Adapter, 3 Optional Intel 10/100 Fast Ethernet Adapters (up to 4) @ $69/ea add'l, PS/2 Keyboard & Mouse/USB/Serial/Parallel Ports, 1U Rackmount Aluminum Chassis (2 HD, 1 CD/FD/HD Drive Bays), 200 Watt 1U PS, Einux Compact Cooling System, Red Hat Linux 7 OS, Also supports Unix, Windows NT/2000 OS. Dimensions: 1.75"H x 19"W x 23"D

Mega features at a mini price!
- 1 to 2 Pentium III CPUs, up to 1GHz
- 256MB ECC RAM, Max 4GB
- 18GB U160 SCSI hot-swap HD
- Up to 2 hot-swappable drive bays
- Built-in Adaptec Ultra 160 SCSI
- Standard with Red Hat Linux 7, also supports Windows NT/2000
- Get Cobalt-like Server Appliance features with optional Browser Manager™ (see sidebar) for $399 more!

T-Rex 1U Dual CPU/SCSI
Pentium III 866 MHz
$2,695 *$135/mo*

Pentium III 1GHz
$2,995 *$152/mo*

Dual Pentium III 866 MHz
$3,195 *$135/mo*

Dual Pentium III 1GHz
$3,495 *$152/mo*

Upgrade to 256MB $99

Upgrade to 36GB SCSI HD $250

Standard Features include Dual CPU Ready Pentium III Motherboard w/Adaptec Ultra 160 SCSI Controller, 2 x Intel 82559-based 10/100 Fast Ethernet Adapter, PS/2 Keyboard & Mouse/USB/Serial/Parallel Ports, 1U Rackmount Chassis (2 hot-swappable SCSI HD, 1 CD/FD/HD Drive Bays), 200 Watt 1U PS, Einux Compact Cooling System, Red Hat Linux 7 OS, Also supports Unix, Windows NT/2000 OS. Dimensions: 1.75"H x 19"W x 23"D

Einux Server Appliances set up in 5 minutes or less!

Einux's optional Server Appliance Browser Manager™ System for Linux lets you set up a Linux-based Einux RackServer in less than 5 minutes using any standard Web browser and a TCP/IP Internet or Network connection. The Browser Manager™ also allows users to access and maintain Einux servers remotely from anywhere in the world. Best of all, the Browser Manager™ lowers the Total Cost of Management (TCM) because it makes it is so easy, almost anyone can set up an Einux Server, thus requiring little or no IT experience. Einux's Browser Manager™ is optional on all Linux-based Einux RackServers and can be added for $399 giving you Cobalt™-like Server Appliance functionality for a fraction of the price! Test drive it yourself at http://www.einux.com/demo.

NAServ Network Attached Storage Filers

Einux's line of Network Attached Storage (NAS) filers let you instantly add storage to your network. Setting up in less than 5 minutes usually, Einux's NAS storage filers let's add storage to your network "on-the-fly" without disturbing any of your mission critical servers. Best of all, they start from 1.1 cents per megabyte giving you cost-effective storage. Test drive it yourself at **http://www.einux.com/nasdemo**

NAS 3000
$1,495
as low as $69 per month[2]
- 40GB[3] Capacity, Max 328GB[3]
- 2 x 20GB IDE Hard Drive
- 128MB Memory
- RAID 0/1/5 Striping/Mirroring/Parity
- JBOD Support
- 10/100 RJ45 Ethernet Port
- 1U Rackmount (1.75"H/19"W/23"D)

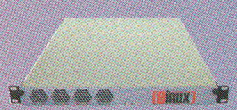
80GB[3] (2 x 40)	**$1,795**
120GB[3] (2 x 60)	**$1,995**
248GB[3] (4 x 62)	**$2,695**
328GB[3] (4 x 82)	**$2,995**

NAS 4000
$2,795
as low as $94 per month[2]
- 160GB[3] Capacity, Max 492GB[3]
- 4 x 40GB IDE Hard Drive
- 256MB Memory
- RAID 0/1/5 Striping/Mirroring/Parity & JBOD
- Optional Hot-Swap Drive Bays (Max 4 Drives)
- 2 x 10/100 RJ45 Ethernet Ports (up to 4)
- 1U Rackmount (1.75"H/19"W/23"D)

248GB[3] (4 x 62)	**$3,495**
328GB[3] (4 x 82)	**$3,995**
410GB[3] (5 x 82)	**$4,995**
492GB[3] (6 x 82)	**$5,995**

Fit Over 20TBs[3] in a 19" Rack

NAS "Plug'nPlay" Instant Storage Solution!



Einux's NAServ Storage Appliances offer "on the fly" storage that can be attached to your network and up in 5 minutes or less. Plus, Einux's NAServs offer the highest density NAS storage appliances in the industry with up to 20.6 Terabytes of storage capacity in a 19" rack.

- 1U Rack height (1.75"H)
- Fit up to 20TBs[3] in a Rack
- From 1.1 cents per Megabyte
- RAID 0/1/5 & JBOD Support
- Easy-to-use NAS Manager sets up in less than 15 minutes
- Easily manage user and group access and security
- Auto detects and configures to your network
- Supports Windows, Unix, Linux & Mac clients
- Remotely manage over Internet with standard web browser
- Optional hot-swappable drives (Model 4000)

Fit up to 168 Servers per Rack!

World's First 4-in-1U and 2-in-1U Server Arrays

Introducing the E.Array, the world's first server arrays based on Einux's patent-pending High Density Cluster Server Array Architecture. Up to 168 individual servers can be installed in a 19" rack making the E.Arrays perfect for IDCs, ISPs and e-businesses where co-location real estate is precious.

- 1U Rack height (1.75"H)
- Fit up to 168 Servers (1/4U) or 84 Servers (1/2U) in a Rack
- Supports up to 850MHz CPU
- Easy-to-use Server Appliance sets up in less than 15 minutes
- Remotely manage over Internet with standard web browser
- Standard with Linux, also supports Unix & Windows NT/2000
- Optional ClusterManager for easy high-availability clustering

Standard Features include 4 x Pentium III Motherboard, 4 x Integrated 10/100 Fast Ethernet Adapter, Built-in SVGA Graphics, PS/2 Keyboard & Mouse, Serial Ports, Einux Proprietary 4-in-1U Rackmount Aluminum Chassis, 1U Power Supply, Einux Compact Cooling System, Red Hat Linux 7.0 OS w/Einux Browser Server Manager, Also supports Unix, Solaris, Windows NT/2000 OS. Dimensions: 1.75"H x 19"W x 27.5"D

1/4U Server Array
$4,995
as low as $175 per month[2]
Model 14633-106
- 4 x Intel Celeron 633 CPU
- 4 x 128MB Memory
- 4 x 10GB IDE Hard Drive
- 4 x Integrated 10/100 Fast Ethernet NIC
- Red Hat Linux OS / Einux Browser Manager

4 x Pent 650 14650-106	**$5,495**
4 x Pent 700 14700-106	**$5,995**
4 x Pent 750 14750-106	**$6,495**
4 x Pent 800 14800-106	**$6,995**
Upg to 4 x 256MB	**Add $599**
Upg to 4 x 20GB HD	**Add $499**

Custom configure your own server at www.einux.com

NETWORK SOLUTIONS
[einux]
serving the next generation

CALL 866.88.EINUX (883.4689)

email: info@einux.com

Toll-Free: 1.866.883.4689 Tel: 626.968.2686 Fax: 626.968.6565

VARs/Resellers: Visit www.einux.com/reseller

[1]Certain models only. Specs may vary. See specific model for details. [2]Leasing figures are based on 36 months an are based on approved credit and may vary. [3]Storage capacities represent raw capacity and may vary according to configuration and partitioning. Not responsible for typo errors. Prices, configurations and availability subject to change without notice. Call for details. All trademarks are acknowledged.

LM0105

er called the "Domain Master Browser" (DMB). The DMB registers its name with the WINS server. All of the local Master Browsers look for this name and will send updates to the DMB, which then combines the lists and hands them back. The DMB is a work-around for the missing NBDD, essentially allowing browsing to cross subnet boundaries.

Liberated CIFS

NBT can be a pain in the neck, and old mistakes still haunt modern implementations. What is a multi-billion dollar corporation to do?

As mentioned earlier, Microsoft introduced SMB without

Open Source CIFS Projects

SAMBA (www.samba.org) is the best-known and most popular Open Source implementation of SMB/CIFS, but there are other projects aimed at leveraging these protocols. This is a partial listing:

SMBFS for Linux

Originated by Pal-Kristian Engstad, SMBFS allows a Linux system to mount SMB shares (shared directories). SMBFS is officially part of the Linux kernel. The project is now maintained by Urban Widmark after changing hands several times.

http://www.samba.org, http://www.kernel.org

SAMBA Appliance Branch

SAMBA Appliance Branch is a slightly stripped-down version of SAMBA intended for Server Appliance devices (headless computers, rack-mounted in server rooms, requiring minimal management). Dave O'Neill maintains this code branch.

http://www.samba.org

libsmbclient

Caldera asked Richard Sharpe of the SAMBA Team to produce an SMB client library for Linux/Unix. When finished, the library will make it easy to add CIFS client capabilities to applications such as KDE's Konqueror. The library is being derived from SAMBA source.

http://www.samba.org

Filesys::SmbClient

Alain Barbet has developed Perl modules that interface to SAMBA command-line utilities and to Richard Sharpe's client library.

http://search.cpan.org/search?mode=doc&query=samba

Implementing CIFS

Yours Truly is desperately trying to document the workings of NBT, SMB, and CIFS to make it easier to implement.

http://www.ubiqx.org/cifs

jCIFS

jCIFS is a set of Java classes that implement SMB/CIFS protocols and is aimed at the development of client applications. Michael Allen heads the coding effort.

http://jcifs.samba.org

SAMBA-TNG

Luke Kenneth Casson Leighton, eager to explore the depths of Microsoft's Remote Procedure Call (MS-RPC) system, created this SAMBA spin-off project with the help of several folks from the SAMBA-Technical mailing list.

http://www.samba-tng.org

libsmb++

libssmb++ is an SMB client library written in C++. Nicolas Brodu started the project, but is not able to pursue it further. This project is looking for a new owner.

http://libsmb.sourceforge.net

Sharity-Light

Based on Linux SMBFS, Sharity-Light runs in user-space instead of as part of the kernel. Note that Sharity-Light is Open Source, but Sharity (no Light) is a commercial product. (Does that make it Sharity Dark?) Both are from a company called Objective Development.

http://www.obdev.at/Products/shlight.html

SAMBA and SMBFS for Amiga

It's not dead yet! Olaf Barthel has ported both SAMBA and Sharity-Light to the Amiga platform.

http://www.amigasamba.org

There are probably more. The Open Source world is a dynamic place with new stuff popping up all the time. Andrew Tridgell, originator of SAMBA, believes in competition within the community and has encouraged more than one of the above projects. Simply put, he believes the more people who know how different projects work, the better.

In a speech at the LinuxWorld Expo in New York, Tridgell said, "Let's just get rid of these horrible protocols." His hope — one shared by many SMB/CIFS developers — is that people will eventually understand how terrible CIFS is, and they will make it go away.

NBT in Windows 2000 and calls it CIFS. In CIFS, the SMB packets run native over TCP without the need for NetBIOS framing. Not only has NetBIOS been removed, but all of the supporting systems (like name resolution, browsing, and even authentication) have been replaced with standards-based services. WINS, for example, has been replaced by Dynamic DNS, and Kerberos is now used for authentication. At the core of all this is the Active Directory, which (like Novell's NDS) is based on X.500. Unfortunately, Microsoft seems to have added their own spin to these services, and several sources are complaining about incompatibilities.

SAMBA can work with Windows 2000 systems, as long as the latter are running in NBT-compatibility mode. Even so, a number of problems have been reported and, hopefully, accommodated. SAMBA lives by adjusting itself to the quirks of each new Microsoft OS.

The Linux community has only begun to dig into Windows 2000. It is an enormous animal that will take some time to dissect. Meanwhile, Microsoft is already working on their next big products. We can only wait to see what changes will come with those.

CIFS and Linux

As we mentioned at the beginning of this article, SMB and CIFS are here to stay. For this reason, Linux must put itself on an equal playing field with Microsoft products if it is to be accepted as a mainstream operating system.

Existing Linux client tools are command-line based, designed for use by folks who already know how this stuff works. Fortunately, efforts are underway to develop CIFS client libraries and other tools aimed at making CIFS easy to use from Linux. These tools will be incorporated into popular desktop systems (such as KDE) so that Linux users can also browse CIFS shares from the desktop…just like Windows users can.

While Linux is working to catch up, Windows has already moved beyond the desktop and into palmtops, settops, and other markets. They have brought CIFS with them. Though it fits well in the back room with the big iron, SAMBA is too big for these environments. Linux needs tiny CIFS clients for the embedded market, simple servers for small-end network appliances, and graphical tools to help it on the desktop.

In the *Open Source Projects CIFS* sidebar you will find information about some of the Open Source projects currently aiming to fully leverage CIFS for Linux and other operating systems. Get involved and get to know what these projects offer. These are the foundation of the future success of Linux in the Network Neighborhood.

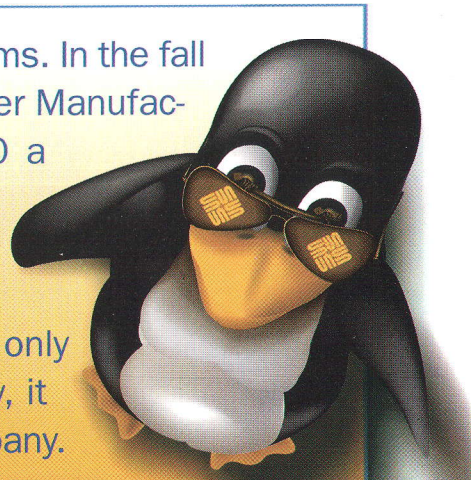
Chris Hertel is a member of the SAMBA Team and a founding member of the jCIFS Team. He can be reached at crh@samba.org.

Sharing the DOT in Dot-Com

Sun Microsystems is learning to live with Open Source whether they like it or not.

by **Robert McMillan**

It's been an interesting couple of years for Sun Microsystems. In the fall of 1999 Java was slated to become a European Computer Manufacturers Association (ECMA) standard, Solaris cost $700 a pop, and the company's Chief Scientist, Bill Joy, was telling *Linux Magazine* that the GNU General Public License (GPL), "just doesn't solve my business problems." A year later Solaris was free, Java was not, and Sun had not only released the largest single whack of GPL code in history, it had also coughed up $1.3 billion to acquire a Linux company.



What Happened?

Conventional wisdom credits Sun with winning the Unix wars by virtue of its most significant and intangible asset — something financial analysts call "focus." Throughout the last half of the 1990s, Sun Microsystems prided itself on having the simplest product line of the Unix vendors. While competitors like HP and IBM struggled with their multi-OS, multi-hardware strategies, Sun became the platform of choice for the new generation of Internet-enabled applications, branding itself as the "dot in dot-com." The strategy worked, in part, because inside that dot was a single microprocessor platform, the UltraSPARC, and a single operating system, Solaris. Focus.

However, focus will only get you so far in the hi-tech industry, and in September 2000 Sun surprised everyone by announcing plans to acquire server appliance manufacturer Cobalt Systems. Suddenly the dot had some company. Cobalt's product line featured two important technologies that were new to Sun — a new microprocessor from AMD and a new operating system called Linux. "It was, in a sense, an admission of failure on the part of their hardware design team," says IDC Program Vice President Dan Kusnetzky. Sun's low-end Netras, with their proprietary architecture, simply could not compete in the high-volume and rapidly growing Web appliance space.

A House Divided

Even before it acquired Cobalt, Sun vacillated between being dismissive (Sun CEO Scott McNealy once called Linux a "better NT than NT") and friendly toward the Linux community. In its new role as a Linux vendor, Sun has a long way to go in convincing Linux developers that it actually wants to help. "They're not seen as contributing anything of value," says SAMBA developer and VA Linux employee Jeremy Allison. He accuses Sun of holding strategic technologies such as Java close to its chest. Not that Sun's reluctance to open-source the popular platform surprises him. "If you're doing well, why would you bother?" he asks.

The list of Sun's sins against Linux is long. Sun is not seen as having given serious support to Linux on its UltraSPARC hardware; its restrictive licensing policy delayed the development of a Java virtual machine for Linux, and it has snubbed key open source developers. In late 1999, the hackers working on the Blackdown project were up in arms when Sun released a Java virtual machine for Linux that included their code but failed to give them credit for their work. Nearly a year later, noted device driver author Donald Becker saw red when Sun released a cross-compiler for Solaris that included his code and was, in Becker's opinion, in violation of the GPL. Add to this McNealy's occasional

Sun Source

Sun is getting its feet wet in the open source ocean. These are the projects it's working on.

Project:	Tomcat
Description:	Reference implementation of Java Server Pages for Apache
URL:	http://jakarta.apache.org/tomcat/index.html
Project:	Netbeans
Description:	Java IDE for Solaris, Linux, and Windows
URL:	http://Netbeans.org
Project:	Open Office
Description:	GPLed suite of office productivity applications
URL:	http://openoffice.org
Project:	GNOME
Description:	Sun plans to replace Solaris's Common Desktop Environment (CDE) with GNOME and is actively working on the open source project
URL:	http://gnome.org
Project:	NFS
Description:	Sun's Unix-standard Network Filesystem
URL:	http://www.citi.umich.edu/projects/nfsv4
Project:	Jxta
Description:	Sun's forthcoming peer-to-peer platform
URL:	http://jxta.org
Project:	Crimson
Description:	A Java XML parser
URL:	http://xml.apache.org/crimson/index.html
Project:	Brazil
Description:	A framework for connecting a variety of devices on the Web
URL:	http://www.sun.com/research/brazil
Project:	Xi18n
Description:	Internationalization enhancements for the X Window System
URL:	http://ftp.x.org/contrib/i18n

inflammatory remarks, and the Linux community's nagging suspicion that Sun sees Linux as more of a threat than it lets on, and you get the basis of a stormy relationship.

But there is no doubt that open source has Sun's attention. Pat Sueltz, the Executive Vice President of Sun's Software Systems Group and the person who is ultimately responsible for both Solaris and Java, says that Sun clearly has a lot to learn from the open source community. "I can see that it actually changes the way we're developing code," she says. "There's an iterative process that comes out of the open source community, and we're exploring that." (See the *Q&A with Pat Sueltz* sidebar, pg. 42.)

And though Sun has spent years casting itself as a white knight in a battle with proprietary software's most vilified dark lord, Microsoft, there is a clear financial incentive for the company to get on the open source train. In a recent report entitled "Open Source Infrastructure: A Manifesto for the Coming 'Big Bang,'" Deutsche Banc Alex Brown analyst Phil Rueppel cited Sun Microsystems as his top pick for investors looking to buy stock in an established company that stood to profit from the open source space. "Despite the company's appearance as being hostile to open source," wrote Rueppel, "we believe that Sun understands both the advantages of an open source development model... and the rich value proposition of a 'single stack' IT solution."

I was young and dumb... It was about 14 months of my life — meeting with people, talking with people.

— James Duncan Davidson, Sun Microsystems, on getting the Tomcat code released.

Rueppel, now a principal with TA Associates in Boston, says that while Sun stands to gain much from open source, it still needs to figure out how to better embrace the Linux and open source communities. "On one hand," he observes, "they don't believe that Linux will work at the high end, so they dismiss it. But on the other hand, they support it because it undermines Microsoft. They're speaking out of both sides of their mouths, and I think that's going to have to be clarified."

Letting Tomcat Out of the Bag

Work on what would eventually become the first significant piece of open source code written and maintained by Sun engineers began in 1998, when a developer inside of Sun's Java group named James Duncan Davidson began a rewrite of Sun's Java Servlet Reference Implementation. At the time, Sun was beginning to get pressure from its Internet Service Provider (ISP) customers who were looking for an alternative to Microsoft's Active Server Pages (ASP) for displaying dynamic pages on the Web. Sun had a Java-based alternative to ASP called Java Server Pages (JSP). The problem was that the Web server code needed to run JSPs was not as mature or as widely deployed as it needed to be and the Apache project was working on its own, divergent implementation of a JSP server. Davidson, who had worked with open source tools before coming to Sun, knew that an open source implementation that was supported by the Apache group would go a long way to solving both of Sun's problems. So, he began rewriting Sun's Java Web server with a view to open-sourcing the code.

There were two problems. No one in management had signed off on the idea, and Sun had never released open source code on this scale. "I was young and dumb," remembers Davidson. "We were really coming out of left field."

Over the next year, Davidson continued to work on the code, but more and more of his time was spent with meetings: QA, standards, and conformance teams, lawyers, and management. Everyone had to be brought on board; that required a lot of open source advocacy. "It was about 14 months of my life — meeting with people, talking with people," he says.

In the end, Davidson says, it was pressure from IBM that

January 1999
Sun releases SCSL, its not-quite-open-source license for Java.

June 1999
Tomcat code released to the Apache Software Foundation.

October 1999
Sun announces plans to release Solaris under SCSL.

December 1999
A press release announcing the release of the latest Java platform for Linux credits Inprise, but not the open source Blackdown project, for its work on the port.

February 2000
Sun creates an open source license — the Sun Industry Standards Source License; an implementation of NFS is the first code released under the new license.

helped get the go-ahead for the project. The help came from an IBM product manager named James Barry, who had been influential in getting IBM involved in the Apache project. Barry was convinced that Davidson's code — Tomcat — would be best off as part of Apache, so he started making some phone calls to Sun. Initially, Sun's management had visions of releasing Tomcat under Sun's Community Source License, but in June of 1999 — after a meeting between Barry and the Java group's General Manager Jon Kannegaard — Tomcat was turned over to the Apache Software Foundation. For the first time ever, Sun was actively supporting an open source project based on their own code.

Blame Microsoft

Sun and Microsoft have never had good relations. In the 1990s Sun saw its bread-and-butter workstation business dissolve in a sea of lower-cost NT desktops, but its most public disagreements with Redmond have been over Java. There has been acrimony, mud-slinging, and massive PR maneuvering since March 1996, when Microsoft signed its licensing deal with Sun, until early this year, when the two companies finally reached an out of court settlement, and Microsoft terminated the agreement.

Though Sun has opened up the way that developers can define the Java APIs, it has remained intractable in its desire to control how the Java virtual machine is implemented. Microsoft says this control has curbed innovation. Sun claims that Microsoft's only goal has been to break Java compatibility. To those who were there, it is not surprising that, after five years under the withering effects of direct competition with Microsoft, Sun would be reluctant to open-source Java. "Microsoft was vicious, aggressive, and really detestable in the way they handled themselves with respect to Java," says Rick Ross, President of JavaLobby, a developer advocacy group that was closely involved in the Sun-Microsoft Java wars. "I think Microsoft really intentionally hurt Java — and hurt it badly," he adds.

On one hand, Sun doesn't believe that Linux will work at the high end, so they dismiss it. But on the other hand, they support it because it undermines Microsoft. They're speaking out of both sides of their mouths, and I think that's going to have to be clarified.

— Phil Rueppel, TA Associates

"The people who took the little seed of Java and grew it to where it is, they just about killed themselves in the process," says James Davidson's current boss, Danese Cooper, who runs Sun's Community Development Program Office. But these days, the old guard is leaving the fold. Alan Baratz — who was in charge of Sun's Java division during the heyday of the Microsoft conflict — left in 1999 to work for a New York financial firm (he subsequently returned to Silicon Valley to become CEO of a startup named Zaplet). Jon Kannegaard — Baratz's interim successor — recently moved to the greener pastures of SunLabs, along with Java's inventor, James Gosling, and former VP of Java Technology and Architecture, Jim Mitchell. "There has been a big shift in personnel," says Cooper.

As the Java warriors leave, new blood, like Tomcat's Davidson, is coming to Sun with new ideas about the role of open source. Pat Sueltz has a pretty good idea of what it means to be a partner in Sun's Java community. She came to Sun after spearheading IBM's Java initiative. Another influential outsider is Marco Boerries, who came to Sun as part of its StarDivision acquisition in 1999. He's the guy who convinced Sun's President, Ed Zander, to GPL the StarOffice code.

Finally, there is Roman Stanek. After his company, Netbeans, was acquired by Sun in 1999, Stanek successfully lobbied to have his core product released under a Mozilla-like open source license. That project, Netbeans.org, became Sun's next foray into the world of open source.

Netbeans project launched.
June 2000

Sun uses Donald Becker's GPLed code in a Solaris cross-compiler, despite Becker's objections; the software is later withdrawn from Sun's Web site.
August 2000

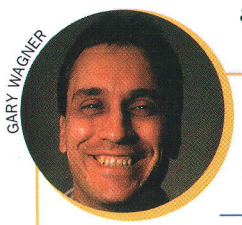
OpenOffice launched.
October 2000

August 2000
Sun announces plans to replace CDE with GNOME and joins the GNOME Foundation.

February 2001
Sun takes on Microsoft.net, announcing plans to release its Jxta peer-to-peer software under an open source license.

Magic Beans?

Sun has long understood that one of Microsoft's greatest strengths is the popularity of its Visual Basic development tools, and since the early days of Java, it has striven to create a market-standard development tool for Java. Rick Ross says that while developers have looked to Sun to produce the Java equivalent of Visual Basic, Sun has failed to produce a contender. "As much as people lambaste it, Visual Basic has been an amazing product at the market level — selling millions of units and used by millions of developers on a daily basis. It's something that can be extended in a thousand different directions by a thousand different modular reusable component products," says Ross. "I don't think any Java development environment I'm aware of approaches its really amazing ease."


GARY WAGNER

"While IBM's engineers are part of the community, Sun engineers are not."

— Jeremy Allison, SAMBA developer

Sun spent five years working on an integrated development environment (IDE) called Java Workshop, but the product never caught on. Having failed to create a VB-like hit with Java WorkShop, Sun began to shop around, hoping to build a popular product out of proven technologies. Eventually, Sun spent $700 million acquiring two tool companies — Forte and Netbeans. The company then proceeded to consolidate the Workshop, Netbeans, and Forte technologies into one product line. "By the time we were capable of executing on a plan, we were like, 'Oh my gosh, we're really late to the party,'" remembers Stanz Kleinen, General Manager of Sun's Forte development tools group. This sense of urgency made Sun more receptive to Stanek's plan to open-source his company's core product. " I think it helped us be more bold and do some things that maybe we would not have done had we had our act together a couple of years sooner," says Kleinen.

The Netbeans project has been growing steadily since Sun — with the help of open source consulting firm Collab.net (which just happens to be the current employer of Tomcat advocate and ex-IBMer James Barry) — released the Netbeans code in June 2000, fostering a small, but thriving, community. According to Sun, there are fewer than 10 developers — outside of Sun employees — who have CVS commit access (the right to change the Netbeans source code), and the project's main discussion list pulls a respectable 40 or so posts a day. Sun also claims to have about 40 partners who are building Java components on top of Netbeans — a decent start for a project that has yet to celebrate

Q&A with Pat Sueltz

If anyone will open source Java, it's Pat Sueltz. After earning a high profile in the Java community as the executive behind IBM's Java efforts, Sueltz came to Sun to head up its software division. She is responsible for Java, Solaris, and all of Sun's open source efforts.

Linux Magazine: Do you think there's space for Solaris in the Cobalt product line?

Pat Sueltz: I have to tell you right now that I'm not looking at Solaris on the Cobalt line because I look at Solaris as the uninterruptible power supply of the Internet. It's customizable and it's rocket science that we've got out there. We've made it much simpler for sysadmins to work with, but it's very different from Cobalt. That's really picking up the gateway, the edge of the network. If people do want to work on the edge, and they want to program to it, then we've got the Netra, and that has Solaris in it.

LM: Are you saying that you don't really see Solaris competing with Linux in the foreseeable future?

PS: What I see is that Linux has enlarged the community of Unix-savvy, Solaris-ready professionals. I have three points I would make about Linux. First, we work very closely with the Linux community as a channel. Red Hat applications port to Solaris readily — there is one-to-one compatibility. We work with J2SE [Java2 Standard Edition] on the Linux environment, so we engage the Linux community as a channel in this way. We have recently come out with J2EE [Java2 Enterprise Edition] for the Linux environment.

The second point is that Linux is not only a channel, it's a community. We're not only working with the major distributors, we've got tools for Linux and we're going to continue to put StarOffice out on all the major systems. So you've got Linux as a channel and Linux as a community, and we participate in both of those.

Finally, there's Linux as an Intel architecture. We're a SPARC-architected company, and I don't play in that game. But the fact is — as you've seen with our appliance strategy — when it doesn't matter what's inside, I don't care. I don't really care what's inside the phones either.

LM: Do you think that Solaris will ever be released as an open source operating system?

PS: I'm not ready to speculate on that yet. I'll just tell you that we continue to contribute mightily to the community. We just made a great contribution with NFS and we'll continue to do those things.

LM: Do you think Java will ever be released under an open source license?

PS: Let me tell you the strategy that I've been working on with Java. It's always been about compatibility. Because it has been made the holy grail of interoperability of applications, there were folks who wanted to co-opt it. Because of that, the path that I've been on is to make sure that I open up the process for changes. So one of the strategies that I came to Sun with was to fix the Java community process. And we've done that. We've now got two bodies of 16 stakeholders; it isn't the entire open community, but it's folks who have licensed Java and includes folks like Nokia, Fujitsu, IBM, BEA, and Apple. We've got the key players that are working on Java as the executive committee to make the decisions about what the future of Java is. And that process is one vote per company. Sun chairs it, but to be sure, the votes don't all go Sun's way.

LM: So you're saying you think Java is basically open enough as it is right now?

PS: You know, I never say never; I will tell you that I think we've opened up the process and that I'll continue to look and see what happens in terms of the open community. This is all an evolution and it's all about ensuring that we stay in touch with the developer community. Right now, I think I've got the process open, and certainly my constituents of the 300 licensees are happy with it, and we're going to keep watching and learning from it.

LM: What do you think are the greatest risks presented by open source?

PS: I don't think we know yet. I think we have to listen and then think about consequences. Remember, I live in a commercial world where millions of people have invested in Sun Microsystems. So I just want to make sure that I make good decisions for my shareholders at the same time that I'm making great decisions for customers and developers. I don't really see a risk. I just think that it's a learning process.

its first birthday, but a far cry from the kind of popularity Microsoft has built around Visual Basic.

The fact that Netbeans has earned far less attention than Sun's best-known open source project, OpenOffice, was intentional, says Sun's Danese Cooper, so as not to turn off the developer community with an avalanche of marketing hype. "The engineers really wanted it to have an engineering focus. They didn't want a lot of hype," she says.

Desktop Duty

With OpenOffice, on the other hand, Sun wanted to get the attention of end users, not just developers, with Sun's PR machine in full gear (news of the project was widely reported as a possible Microsoft-killing move on the part of Sun). It was followed, a month later, by news that Sun planned to drop Solaris's standard desktop software, CDE (the Common Desktop Environment), in favor of an open-source alternative, GNOME — an announcement that was, again (surprise, surprise) played up as a possible threat to Microsoft. (For more on GNOME and OpenOffice, see *Linux Magazine's* December story, "The Meaning of GNOME," located on the Web at http://www.linux-mag.com/2000-12/gnome_01.html.)

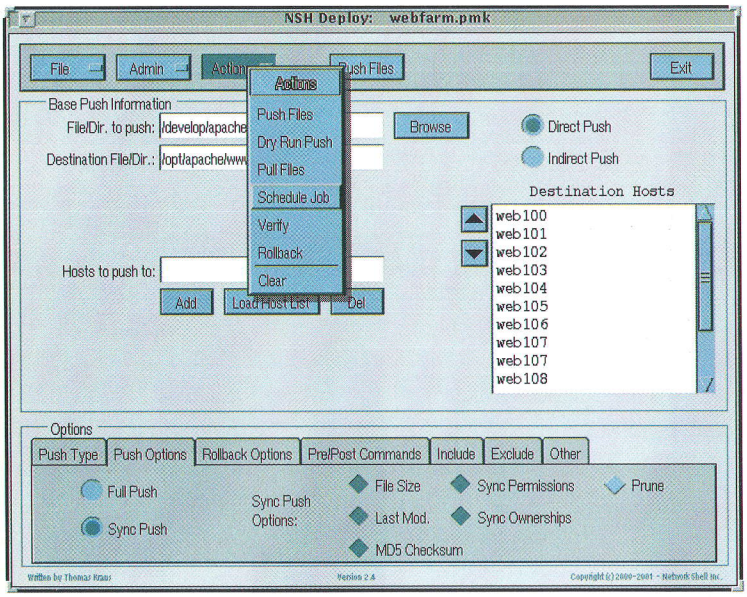
The OpenOffice project has yet to gain serious developer momentum, but in December 2000 — two months after its launch — Sun passed an important milestone by giving the first non-Sun developer the right to modify the OpenOffice source code. And Sun's GNOME efforts appear to be on the right track. "It's certainly going a lot better than a lot of other corporate involvement that we've seen," says Havoc Pennington, the chair of the GNOME Foundation Board. "We're getting people involved in the discussions; we're getting patches back."

I can see that [open source] actually changes the way we are developing code. There's an iterative process that comes out of the open source community, and we're exploring that."

— Pat Sueltz, Sun Microsystems

Still, it's not clear how strategically important these desktop-focused initiatives are to either Sun or Linux. Sun itself has a hard time articulating how a company whose fastest growth is now occurring in data centers is going to profit from releasing a free, GPLed office suite. Bill Roth, the Group Product Manager for OpenOffice, says that the project is, in part, an attempt to win the hearts and minds of open source developers. "We had to make a kind of Nixon-going-to-China

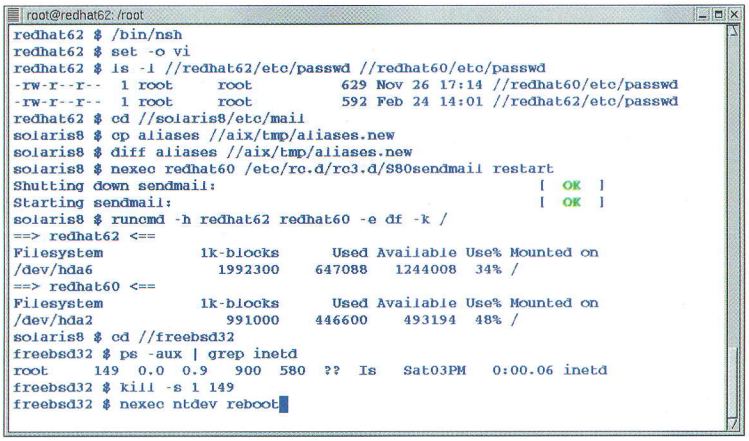
NSH Deploy



NSH Deploy Features/Benefits:

- Massively parallel pushing for minimal deployment time
- Job scheduling for automated deployment
- Pull files as well as push files for extended functionality
- ACLs and encryption for maximum security
- Independent verification for *anytime* consistency checking

NSH Utilities



The Network Shell (NSH) Utilities are a UNIX *shell* environment with distributed capabilities. In other words, a distributed scripting language giving you secure access to multiple UNIX, Linux, and/or Windows servers at once from a command line, or (shell) script, without the need to ftp, telnet, or rlogin to each server one at a time.

- Manipulate files, directories, and processes
- Execute remote commands and scripts
- Encryption, compression, and ACLs.

For a fully functional evaluation please visit or contact

WWW.NETWORKSHELL.COM

1 888 I WANT NSH or 650 548-0674

The Myths of Open Source

In pitching the open source philosophy to their peers at Sun Microsystems, the Netbeans team came up against four commonly held myths about open source:

Myth 1 Anybody can change code on your project's CVS at anytime.

Myth 2 Open source means free labor.

Myth 3 Open-source it and they will code. "This isn't the Chia Pet of software," says Sun's Danese Cooper.

Myth 4 The only smart engineers are the ones employed by your company.

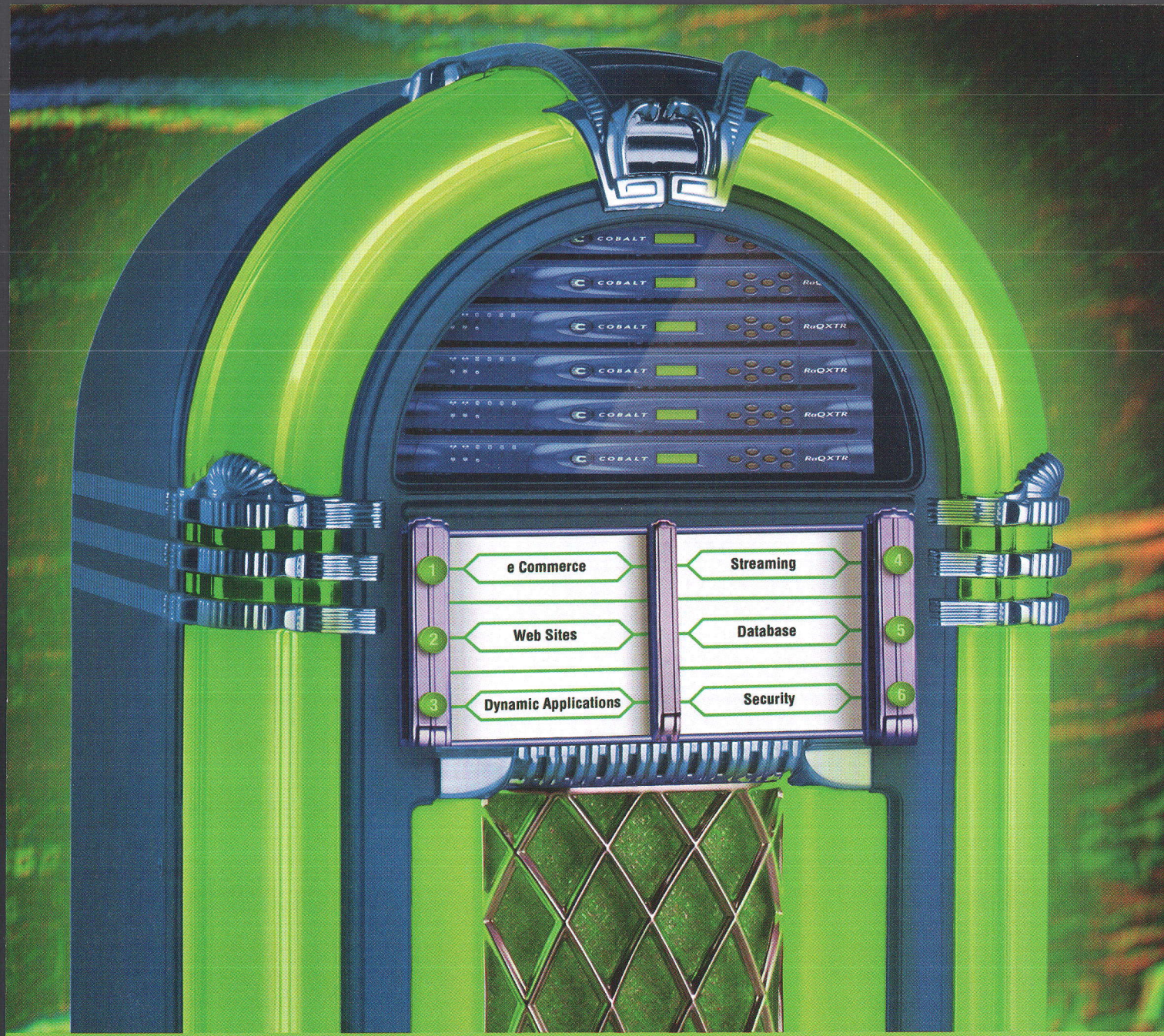
effort to prove to people that we were serious about this."

SAMBA's Jeremy Allison says that if Sun really wants to convince the open source community that it is serious, it should take a page from IBM's playbook. IBM, says Allison, has developers on a wide variety of mailing lists who are actively contributing patches to all parts of the open source community. Allison maintains that Sun engineers tend to focus on Sun-sponsored projects and, with the exception of GNOME, tend to have a pretty low profile. "IBM's engineers are part of the community," he says. "Sun engineers are not."

Respected developer Don Becker, who is still unhappy about the way Sun used his GPLed driver code in a Solaris cross-compiler, agrees. In an e-mail interview, he wrote, "There are parts of Sun that seem reasonable, but most of Sun is aggressively anti-open-source," adding, "I'm very impressed with how IBM has understood Open Source and the GPL as it relates to doing business and working with the community. I'm sending this letter from the Extreme Linux Developers Forum. IBM people gave a talk. No Sun people are here."

All this will change if Danese Cooper gets her way. She says that Sun has not yet reached the mid-way point in its evolution into an open-source-aware company, but successes like Netbeans and Tomcat are going a long way to assuage the trepidation within Sun. In addition, she is spending much of her time traveling around to Sun's different divisions, explaining the open source philosophy and helping Sun's product people to see where it might fit into their future. One of the few people working for Sun who has publicly stated her support for the open-sourcing of Java, Cooper is optimistic that Sun will make the transition into the open source community. "I think that Sun is culturally ready for this change to happen," she says.

Robert McMillan is editor at large with Linux Magazine. *He can be reached at bob@linux-mag.com.*

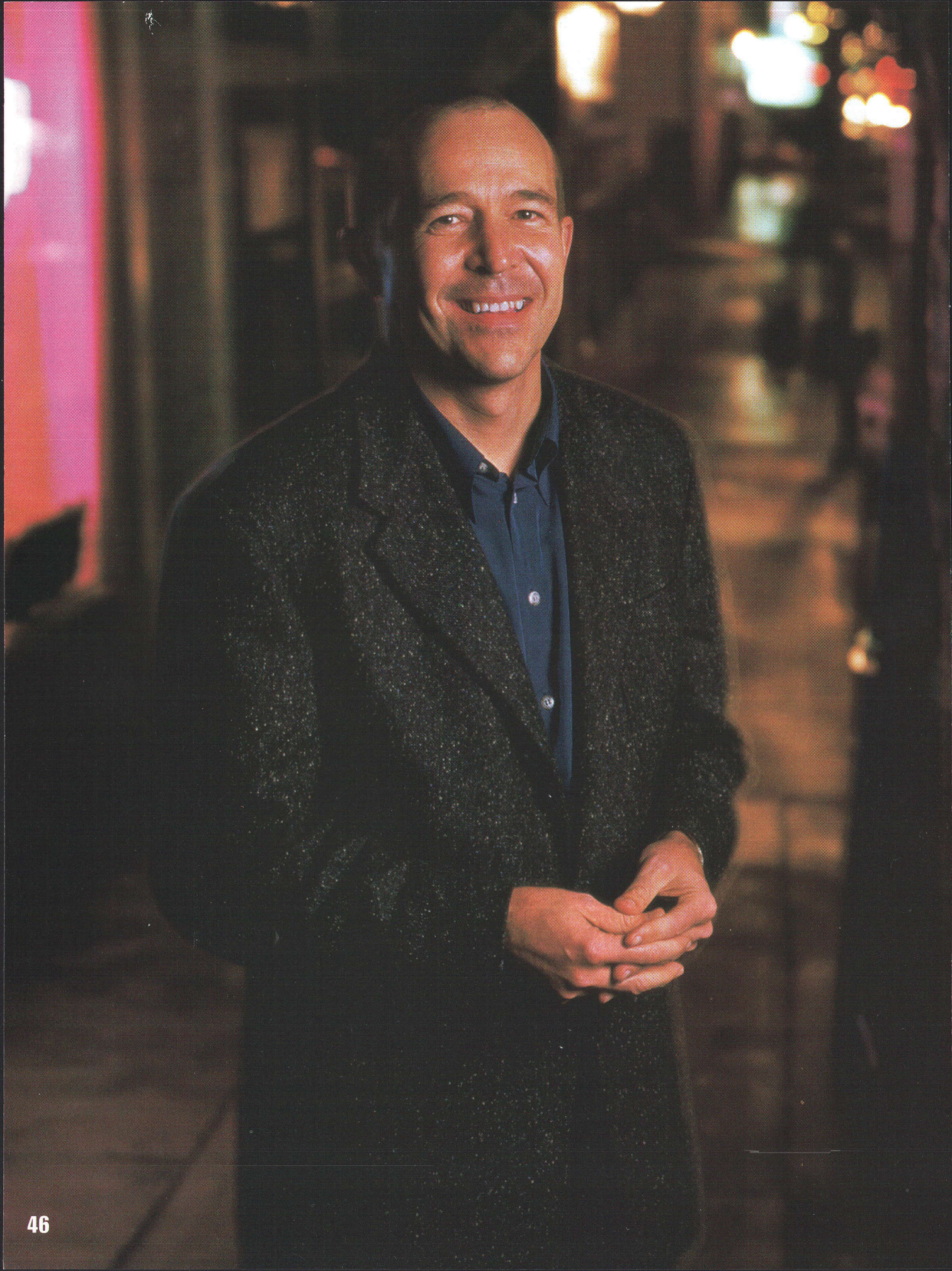


Rock Your App Off!

Sun Cobalt™

Does your application hosting require a little more rock and roll? Tune in to Sun's newest server appliance, The Sun Cobalt™ RaQ XTR server. With four front-panel-accessible RAID drives rolling, a new processor rocking, and show-stopping performance, now you can crank out the hits. Because it's a server appliance, you just unpack it and plug it in. No software or server expertise needed. Ready to race up the charts?

Tune in to www.cobalt.com/linuxmag and get ready to rock your app off.



BUSINESS AS unusual

RED HAT'S MATTHEW SZULIK IS INVENTING THE LINUX BUSINESS MODEL

by Robert McMillan

What makes a successful open source company? If there's one corporation worth examining to answer this question, it's Red Hat. Since awakening the financial community to the open source phenomenon with its remarkable IPO in August of 1999, Red Hat has steadily been writing the book on how to build a Linux business. *Linux Magazine's* Adam Goodman, Robert McMillan, and Chris Somerville recently caught up with Red Hat CEO Matthew Szulik to discuss the lessons learned during his company's first year and a half as a public company. Szulik, a mild-mannered Bostonian who had recently shaved his head for a charity, has seen a fair bit in his first year at the helm: A dozen acquisitions, executive flight (CFO and CTO), and a stock that soared to $300/share and then plummeted to 1/30th of that value. But in the end, he says, it's the future, not the money, that motivates him each day.

Linux Magazine: The world has sobered up since we spoke with [Red Hat Chairman] Bob Young last year. How has it felt to have your stock drop from such a height?

When our stock hit $300 and there was wild enthusiasm, I said, 'This is one of the worst days of my career.'

Matthew Szulik: When our stock hit $300 and there was wild enthusiasm, I remember having this discussion with Bob. I said, "This is one of the worst days of my career." He asked, "Why?!!" He was having a good time and was caught up in all of this.

I had watched a group of incredibly bright young people work their butts off, deal with just incredible changes and challenges, and I thought that a wild stock price — an $18 billion market cap on the strength of a boxed product — was going to take away from all of the good work that had been accomplished, and most importantly, from the bigger challenges that lay ahead of us.

All the Microsoft hoopla made good theater, but Microsoft is a company with $26 billion in cash in the bank. They're an incredible American success story, and with all due...actually with a *lack of* respect for their practices, I knew that the hard work was before us. We had yet to be challenged.

With all of that in mind, I didn't see the wild enthusiasm. I wasn't one of the cheerleaders. I was saying, we've got to get the follow-on offering out; let's move. We've got to get the Cygnus acquisition done; let's go. We've just got to keep scaling and keep driving the company to perform as rapidly as possible.

LM: So if you could do it again, would you go public when you did?

MS: I wouldn't change a thing.

LM: But you thought it was a bad thing when the stock hit 300?

MS: Because of what's happening now. Today we're sitting here and the stock's got a $1.3 billion market cap and there are probably people writing stories that I should be slicing my wrists. When you take a look at the complete picture of who we're choosing to compete against, when you look at our financial position — we've improved the gross margins of the business from 46 percent 16 months ago to 60 percent in this last quarter — we've got $320 million in the bank; we're on the verge of being profitable. We've got great relationships with IBM and Compaq and Dell and others.

Why do people want to make me feel lousy? We have introduced 16 new lines of revenue into our company. We've been able to hire great people. And I think one of the most important elements is that we have been true to our commitment to the open source model. The questions used to be, "When are they going to go to the dark side? When are they going to start adding proprietary layers? When are they going to flip the model and become a proprietary company?" Heck, we cancelled revenue streams that had a proprietary component in them so we could continue to keep this vision and keep this momentum alive.

LM: What revenue streams did you cancel?

MS: When we bought Cygnus, they had a 75-person development effort working on a proprietary IDE (integrated development environment) for the semiconductor simulation market. It was called the Churchill project. We cancelled it.

LM: How did your employees react?

MS: It made a lot of people very angry. People had devoted a year of their lives to developing this technology.

LM: What led you to acquire Cygnus?

MS: Let me describe some of the conventional thinking we were hearing at the time. We were hearing from enterprise customers that in order to be competitive with Microsoft and others, we needed to have a platform with a consistent API [Application Program Interface] from server to handheld device. If you're really going to be embraced strategically by the Global 2,000 — especially as wireless and other technologies loom as part of their five-year IT plans — you need to have an answer to that, and we didn't at the time. So that was issue number one.

Second of all, there was a very popular question a year ago. "What about the desktop? Why don't you go out and buy Corel?" But Microsoft had 92 percent of that category. We didn't think it was a very forward-looking category. A lot of smarter people than us fought that war and lost. As a Bostonian, I had watched Lotus's franchise crumble under the weight of trying to compete with Microsoft. We made a strategic decision at that time to move forward and not compete on the existing playing field. We would get into the handheld and device marketplace. We thought that that was where the next frontier would be. And all of this gets tied together with the Red Hat Network. That is the cornerstone of the whole company.

LM: How is what you're doing with Cygnus different from what you originally expected?

MS: The conventional rule of thumb is that 65 to 70 percent of acquisitions fail. Cygnus was 11 years old when we acquired them and had gone through multiple changes over its history. What we saw was great engineering talent; the core engineering leadership of the

company had a strong commitment to open source. Most importantly, they had great customer references. They knew how to build sophisticated projects and get them delivered on time and on plan. That was a discipline that we thought was important for our company. When we acquired them, we were just completing the follow-on offering and we were also a very young company. So I think we've learned a lot about how to integrate companies. From that perspective it was a great learning experience for us. And we've learned to do better with our subsequent acquisitions.

LM: How has it played out then? Are there any customers that you can talk about that have been able to take advantage of your competencies in this wide range of offerings?

MS: Oh, definitely. I don't think the people that we've partnered with would see us as a complete solution provider if we didn't have an answer to the portable device story. If we didn't have that, we would have given them a reason to move away from the Red Hat relationship, and we wo
po
Wi

LM
the
do

M
ma
pe
sa
ni
be
uc
W
ar
be
th
w
th
is
ho
ti

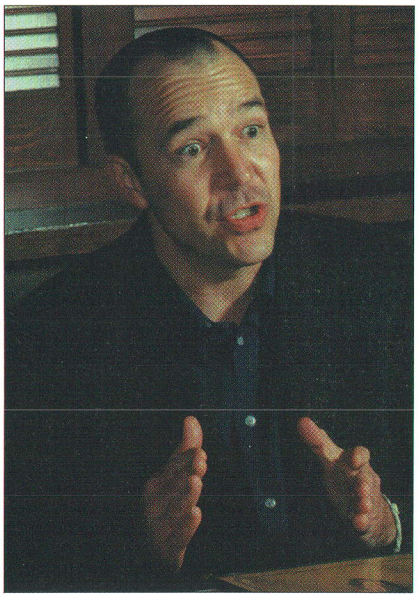
continuous innovation, continuous updates, and do that on a monthly subscription basis."

If you look at that, you'll see that we chose to start a subscription relationship at the expense of perhaps some valuation upside. If you look at our balance sheet, you'll see a line of deferred revenue there of $12–13 million, which unfortunately we don't get a whole lot of credit for from [Wall Street]. This was the beginning of our strategy for getting into the subscription model — to build and monetize a relationship with the millions of people who have touched the Red Hat brand. The Red Hat brand could be expanded to include the distribution of not just our technologies, but other technologies as well.

LM: Bob Young has said that he sees Red Hat as the Wal-Mart of technologies — that you can buy all these open source technologies through Red Hat.

MS: He never said that to me, but I'm glad he thinks that way.

LM: He has a metaphor for almost every occasion.



that will continue to be the case. You'll throw it away, and new functionality will be added. But when we have spoken to executives at Nokia, Ericsson, Sony, and others, their cost of customer acquisition is so high that the more they can enable that device to stay in your hand and use the network to upgrade that, the more they're able to continue and keep that brand relationship with you. It's not the manufacturing cost that they're worried about; it's the cost of switching.

In speaking to Ericsson about the Web Pad, we would like very much to enable the enhancement of that device, or the extension of new services or new value-add to it. We would want to be involved whether it's through Ericsson.com, which is being backended by Red Hat, or directly through Redhat.com, where we can introduce new services and new value-adds and participate in the revenue stream on the incremental dollar amount that Ericsson will be charging their customers.

When we bought Cygnus, they had a 75-person development effort working on a proprietary IDE. It was called the Churchill project. We cancelled it.

LM: Why do you think the subscription model is the right one for Red Hat?

MS: I think two things make subscription work for Red Hat. One is our revenue-recognition practice, because for a company that's been basing their business model and their financial reporting on the sale of software licenses — to go back and restate that as a subscription, especially as a public company, is not easy.

The second is that one of the great benefits of open source is the continuous improvement of the product. So when a customer buys a subscription — if the release is every two to three years like Solaris, or four years like Microsoft — where's the value? In the case of Red Hat, we produce two to three versions every 14 months.

LM: So how fast is the Red Hat Network growing? Can you tell us how many subscribers you have?

MS: I believe when I left the office, we had in excess of two million updates of the product, and I think it's an evolving activity for the company. We have stated publicly that this upcoming fiscal year we don't expect it to have a major impact on our revenue model.

LM: So who is paying for the Red Hat Network now?

MS: It's a free service right now.

LM: How will it make money?

MS: Shortly, through registration. Are we smart enough to create enough value that people would want to pay for that service? This is a strategic move for our business, because it's not simply creating an online service. There will be many other strategic initiatives related to how we will reposition our company to support this. And it's not trivial. When you start telling people that you're going to manage their 5,000 or 10,000 server environment, and that they're going to be able to have a complete snapshot of their entire architecture, and that you're going to maintain it and manage everything from security to remote systems management…I think we're learning a lot.

So we have a number of very large strategic customers that we're working with right now. We've been able to learn a lot and we continue to learn from them how to do this responsibly, making sure we don't shoot ourselves in the foot by over-hyping it.

LM: So you see yourself becoming an application service provider (ASP)?

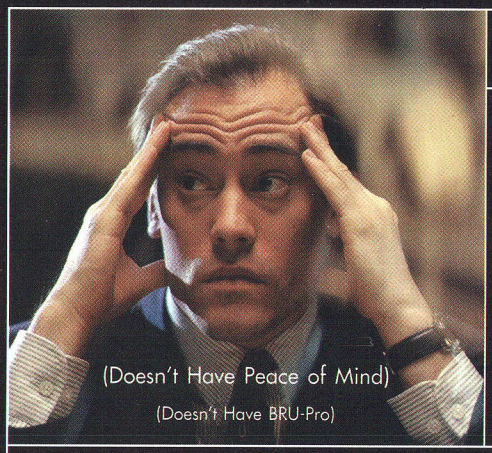
MS: Absolutely.

More than Money-Motivated

LM: We've been talking a lot lately about the relationship between commercial and community interests. I have a hard time coming up with the names of any developers who have come into the Linux community — say from IBM — and become well-known leaders on the basis of their work. Why do you think that is?

MS: Well, I have a funny story. We hired a very, very bright developer — a very smart young man. And he came in after his 90th day at Red Hat and said, "I really believe in the technical direction of open source," but it turned out that even though he had the technical capability to make contributions, he found that access to information — and to the leadership of the project that he was working on — was highly fragmented and it was difficult to get his ideas heard.

It's no different from when you get married and have a child. That represents change. When you have another child, that represents more change. I think change is hard for some people. What was a somewhat quiet, very technically-driven community of people using technology for their own purposes is now mushrooming into a commercial industry. And I think what has to be taken into consideration is; A) What are the long-term benefits of expanding out the community? We need to recognize that the community is going to change. There are social and market forces which will cause



AVAILABLE

For Linux Data Protection You Can Trust, Follow the Leader

BRU PRO™

PROTECTS YOUR ENTERPRISE DATA ASSETS.

The new professional backup and restore solution for your network from Enhanced Software Technologies.

BRU-Pro™ provides cost-effective and reliable Linux system solutions for your tape servers with rock solid functionality at realistic pricing. BRU-Pro includes support for virtually all SCSI tape devices and libraries, eliminating the need to purchase expensive drivers. Take advantage of EST's extensive Linux expertise with cost-effective 24/7 technical support.

For over 16 years Enhanced Software Technologies has continued to be the leader in technology for protecting your data's integrity. We invented the concept of Backup You Can Trust℠ and continue to provide the proven reliability necessary to protect your critical data.

BRU-Pro Features

- Fully verified backups for accurate restores
- Detect and recover from errors on restore
- 16 years of proven data protection experience
- Based on BRU, the #1 award-winning Linux backup product
- Automatic client recognition and configuration
- Automatic tape device recognition and configuration
- Graphical interface eases administration
- Unattended, lights-out backup
- Quick File Access (QFA) shortens restore times
- Automatically recovers from tape errors
- End users may restore their own files
- Offline verifications to accommodate shrinking backup windows
- GUI access provides complete functionality from any location
- Web interface to perform restores
- Administrator-defined security levels

EST™
ENHANCED SOFTWARE TECHNOLOGIES
Software You Can Trust℠

For More Details About BRU Pro visit:

www.estinc.com

From the Developers of the Award Winning BRU™

accesses e-mail on a different computer system from the one where the official mailbox is located (the target location for the delivery agents). The storage location is known as a "message store." The user agent must then connect to the message store in order to view, access, manipulate, and potentially download the messages, using the Post Office Protocol (POP3) or Internet Message Access Protocol (IMAP) for communication.

Figure One (pg. 58) illustrates some of these components and concepts using a sample mail message sent from a user named `Hamlet` at `uwitt.edu` to his friend `Ophelia` at `ophe624@elsinore.gov`.

Hamlet composes his message to Ophelia using a mailer program like *pine* or *mutt* on one of the common workstations in his department (hostname *philo*). Depending on his user agent and its exact configuration, it will either forward the message to the local *sendmail* process using port 587 (allowing *sendmail* to submit it to the mail facility), or it may do the submission operation itself, communicating with *sendmail* via SMTP on port 25 (the transport agent standard port). In our example, *pine* has been configured to function as a submission agent as well as a user agent, while *mutt* relies on *sendmail* for mail submission.

At this site, all outgoing mail is funneled through a single mail relay host named *apollo*, The *sendmail* process on *philo* passes the message along to the corresponding process on *apollo*, which routes it to the Internet. From there, the message will eventually be sent to `ophe624@ elsinore.gov`, which is diverted (via DNS `MX` records) to some system at the ISP used by the `elsinore.gov` site.

When convenient, the incoming mail server at `elsinore.gov` (*poste*) connects to the ISP and uses the *fetchmail* program to retrieve waiting messages. *fetchmail* forwards the data to *sendmail* using the SMTP protocol and port 25, thereby simulating normal incoming TCP/IP mail. The *sendmail* process on *poste* sends the messages for user `ophe624` to the *sendmail* process on *polonius*, where the *procmail* program places it in the correct mailbox on host *polonius* (*/var/spool/mail/ophe624*).

From the point of view of the *sendmail* transport agent, the mail is now delivered. However, `Ophelia` still has not seen the message. She reads her mail on her laptop. To do so, she has configured the mail program in Netscape to connect to the message store (her mailbox) on *polonius*, providing the appropriate username and password for authentication. Netscape will then display information about the messages in her mailbox, showing her the actual message when requested, retrieving all data via the IMAP protocol. `Ophelia` can delete the message, download it to her laptop, or file it away into one of her mail folders on *polonius* (or even leave it in her incoming mailbox).

If the site `elsinore.gov` had a direct Internet connection, the initial delivery of mail message to their site would be somewhat different. Instead of using *fetchmail* to retrieve messages from a remote ISP site, mail would arrive at the site's firewall. In this case, a daemon that forwarded SMTP packets to a designated host inside the firewall (*poste* could again be used for the latter) would run on the firewall. Such a daemon is known as an SMTP proxy; the most widely used SMTP proxy facility is the combination of *smtpd* (to receive and store incoming SMTP data) and *smtpfwdd* (to forward SMTP data to the incoming mail server) from www.obtuse.com.

Mail Address Subtleties

So far, we have considered only the most straightforward mail-addressing scenario — a message is addressed to a user at a particular site. However, some complications can arise:

- DNS `MX` records can redirect messages directed at some host to a different host.
- E-mail aliases can redirect incoming messages for a user to a different host and/or user (or even group of users).
- Mail-forwarding mechanisms can also redirect mail to a different destination address (typically used for users who are away from their home

Mail System Components Available for Linux Systems

User Agents: *mail, elm, mutt, pine*, mh/nmh*, Netscape*, emacs' rmail* (Starred items can also function as submission agents.)

Submission/Transport Agents: *sendmail, Postfix, smail, qmail, exim.*

Delivery Agents: *mail, deliver, mail.local* (part of the *sendmail* package), *procmail, uux* with *rmail* (for UUCP-transported mail) — Mail piped to programs is delivered by */bin/sh* (*bash* actually) or *smrsh* (part of the *sendmail* package).

Access Agents: Any mailer program supporting the POP3 or IMAP protocol (these include *pine, mutt, mh/nmh,* and *Netscape*) — When the message store is kept on a Linux computer system, it will need to run the appropriate daemons; these may be found at http://www.washington.edu/imap. Some Linux distributions also include IMAP and POP daemons in a package with a name beginning with *pop* (followed by a version number string).

Retrieval Agent: *fetchmail* is the program of choice for these functions.

YDL 2.0

by Terra Soft Solutions, Inc.

www.yellowdoglinux.com

Redefining Your PowerPC Experience

site for an extended period or have left an organization altogether).

We will consider each of these in turn.

MX Records

DNS MX records specify the host that handles e-mail for a designated computer system. They cause e-mail to that host to be preempted and rerouted to the new destination system. MX records have the general format:

```
host [ttl] IN MX n destination
```

where host is the computer to which the record applies, n is a number indicating the record priority (lower numbers indicate higher priority), and destination is the name of the host to which mail should be redirected (note that it can be the same as the host itself). (ttl is the usual optional time-to-live parameter applying to DNS caching. In case you need it, you can find a more general overview of DNS on our Web site at http://www.linux-mag.com/2000-09/dns_01.html.) *Figure Two* shows some examples of this.

Host dalton normally receives its own mail since it is listed as its own highest-priority destination host. If dalton is unavailable, mail will be redirected first to host postal and then to host remote.mysite.com. In contrast, e-mail destined for host newton is redirected to host apple under normal circumstances. If apple is unavailable, mail will go to postal instead. Thus, postal serves as a backup mail server for both hosts (and may do so for an entire network).

The final two lines specify a default mail destination system for mail addressed to anyuser@mysite.com; by default, mail will go to the system granada, which serves as the incoming mail server for that site. System laguna is specified as a backup.

Mail Aliasing

Mail aliases are another way of rerouting e-mail. In contrast to DNS MX records, these operate on a per-user basis. Mail aliases are defined in the file */etc/aliases;* this feature is part of the *sendmail* package (and also of other transport agents). Note that some mailer programs also allow personal mail aliases to be defined, but these apply only to outgoing messages and won't be considered here.

Figure Two: Sample MX Records

```
dalton          IN      MX      10      dalton
                IN      MX      20      postal
                IN      MX      90      remote.mysite.com

newton          IN      MX      10      apple
                IN      MX              postal

mysite.com      IN      MX      10      granada.mysite.com
                IN      MX      20      laguna.mysite.com
```

Entries in the aliases file have the following format:

name: user [, user ...]

Here are some example entries:

```
aef:  aefrisch
aefrisch: aefrisch@dalton
max: \maxwell@newton

chem.: aef, chavez, jones
phys: aef, enzo, nadia
science: chem., phys, max
jones: djones@stockmkt.com
```

The first two entries illustrate user account aliases. In this case, mail for aef would be redirected to aefrisch. The name aefrisch is itself an alias and expands to aefrisch@dalton, so mail coming to this system for aef would go to aefrisch@dalton (at least to start with). Note that aliases continue to be expanded as long as is necessary. The third entry defines an alias for max: maxwell@newton. This is a terminal alias; the initial backslash prevents any further expansion.

The second group of sample entries is used to define some mailing lists. The first two lists have three members each. The third list, science, has two other mailing lists as its members (along with max). Any duplicates in the resulting list will automatically be removed by *sendmail.* Note also that entry order is irrelevant in the aliases file. Thus, the alias jones does not need to precede its use in the chem mailing list.

Whenever you edit the aliases file, you must convert the text file to the actual binary databases used by *sendmail* by running the *newaliases* command (equivalent to sendmail -bi). Aliases may also be used to redirect mail to a file or a program; we will not concern ourselves with these capabilities here.

Forwarding Mail

Mail forwarding is the third mail-redirection mechanism we will consider. It is very similar to mail aliasing. There are two main mechanisms for accomplishing it. First, a file named *.forward* may be created in a user's home directory. It should contain one or more e-mail addresses to which e-mail should be forwarded (the simplest format is to put each address on a separate line). For example, if the *.forward* file in user chavez's home directory contained the single line ruiz@umexico.edu, then her e-mail would be forwarded to the specified address. If she wanted to keep a local copy of the mail as well, she could use this *.forward* file:

```
ruiz@umexico.edu
"/home/chavez/mail_pile"
```

This file would forward the mail to

the same address and also place a copy of each forwarded message into the file *mail_pile* in her home directory.

The second mail-forwarding mechanism is an internal feature of *sendmail*, and we will discuss it next month.

Putting It All Together

How do all of these separate pieces interact to affect mail delivery? Basically, at each attempted delivery host, `MX` records are examined to see whether e-mail is rerouted at the DNS level or not. If so, the mail is sent to the same user at the new host, whose own `MX` records are then checked.

If no `MX` record has an effect, then the address is examined for aliasing via the *aliases* file and then the forwarding mechanism. Either of these has the potential to redirect the mail to a different user and/or host. If the host changes, the message is routed to the specified host, where `MX` record-checking occurs again.

If all of the aliasing does not redirect the message to a different host, then the message gets delivered to the final resulting user on the local system.

Another example should be able to make this clearer. Suppose a message is addressed to `jane@doe.com`. It arrives first at the incoming mail server `poffice.doe.com` (designated by the `MX` record for the domain itself). At this point in time, the `MX` records for `poffice` are examined, and its highest-priority record redirects the mail to host `sorter`. Thus, the address now becomes `jane@sorter.doe.com`, and the mail is redirected at the DNS level.

Now suppose `sorter`'s `MX` record points to itself, but an alias on that system defines `jane` as `jane@incognito`. The address now becomes `jane@incognito`, and the mail is transported accordingly. Host `incognito` in turn has an `MX` record pointing its mail to `erewhon`, so the address becomes `jane@erewhon` (and the *sendmail* process on `incognito` never handles the message).

Finally, on `erewhon` (which has an `MX` record that points to itself), `jane` is aliased as `jsmith`. User `jsmith` has a *.forward* file which consists of the entry `kjones@faraway.org`. So, the mail is then readdressed to `kjones@faraway.org`, and the various *sendmail* processes in this domain send it back out onto the Internet. When the message finally arrives at `faraway.org`, the entire process begins again.

More Next Month...

Hopefully, the information in this column has given you a good feeling for what is involved in making an organization's mail system work successfully. Next month, we will consider many of these concepts in more detail as we discuss configuring *sendmail* for various, common uses.

Æleen Frisch is the author of Essential System Administration. *She can be reached at aefrisch@lorentzian.com.*

COMPILE TIME

The Fibers of Threads

By Benjamin Chelf

For the last several months in this column, we've been looking at programming with Linux's threads library, *pthreads*. However, we have taken for granted the work that is actually done under the covers by the *pthreads* libraries. So this month's *Compile Time* will dissect Linux *pthreads* themselves to discover exactly what it is that makes them tick.

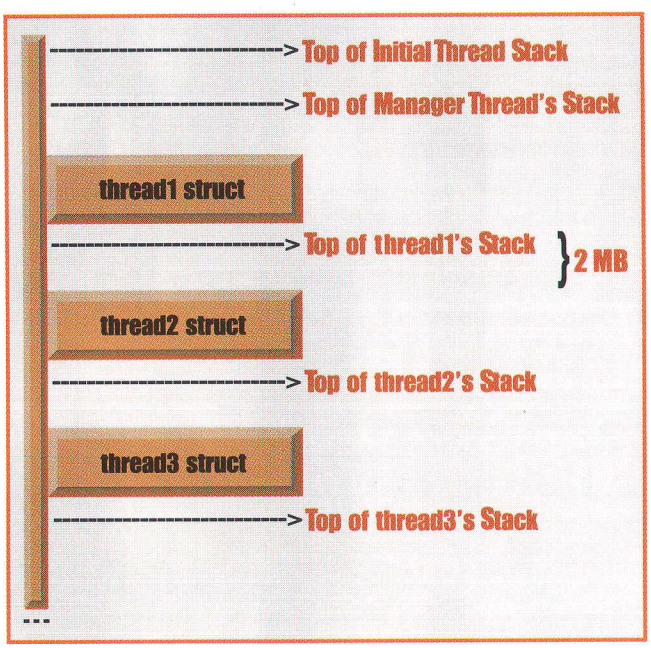


Figure One: A stack after a few threads have been created.

Before we dive in and start looking at this topic though, you'll probably find it a lot easier to follow if you are already familiar with a couple of other concepts. First, we assume that you have an understanding of the difference between code that runs in "privileged mode" (i.e., kernel space) and code that runs in "user mode" (i.e., user space). It's also important that you understand how system calls work and how they switch between user mode and privileged mode. If you need to brush up on these topics, a good introduction can be found in the May 1999 *Compile Time*, located on the Web at: http://www. linux-magg.com/1999-05/compile_ 01.html.

There are three basic ways that threads can be implemented on a system: many-to-one, one-to-one, and many-to-many. Each method requires a different amount of support from the operating system and has its own strengths and weaknesses. The names of the methods refer to the number of threads the application 'sees' in user space compared to the number of threads (i.e., processes) the operating system sees in kernel space.

Many-to-One

In a many-to-one implementation of threads, the threads are completely written in user space. From the kernel's point of view, there is only one process executing. Although that process is multi-threaded, to the kernel it looks the same as any other non-threaded process. The advantage of this method is that it requires no additional support from the kernel and therefore can be implemented on any system. All the thread management issues (including scheduling) are handled in user space by the application.

Interestingly, the many-to-one implementation's greatest strength just happens to be its greatest weakness as well. Because the operating system views the multi-threaded application as a single process, that application can never take advantage of additional processors if they exist. The multi-threaded application is guaranteed to run only on a single processor even if additional resources are available.

In fact, in many cases, the multi-threaded application will not even act like a multi-threaded program at all. For example, if one of the threads is waiting for a system call to return, the entire application must also wait! When a system call executes, the calling process is required to block until it completes. Therefore, in a many-to-one implementation, all threads will stop executing when any one of them needs to switch into kernel space. This is intolerable behavior in a threaded application (e.g., imagine if all the windows in your GUI froze up when the application was waiting for user input!) and, therefore, this implementation is not really a solution to the problem of implementing threads.

One-to-One

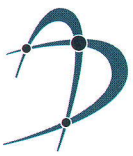
The one-to-one implementation of threads solves the two problems of the many-to-one implementation listed above. In this implementation, every thread in a multi-threaded application is its own process in the kernel. This implemen-

~~High performance~~

~~Extreme~~

Speed-of-light data management solutions

Is your data traveling at the speed-of-light? Because every nanosecond counts, our solutions deliver responsiveness, always-on availability and unlimited scalability for 1,000 users or 1,000,000. All while reducing your time-to-market up to 80%.



Object Design

Discover why Object Design has redefined data management performance for more than 4,000 companies for over 10 years. Jump light-years ahead of your competition, with best-in-class products that include ObjectStore®, Javlin™ and PSE Pro™.

www.objectdesign.com

Figure Two: Prototype for clone()

```
int __clone (int (*fn) (void *arg), void *child_stack, int flags, void *arg)
```

tation is often simpler than the many-to-one implementation because it allows the kernel to perform the scheduling of the threads just as it does for other processes. Also, if the machine on which the multi-threaded application is running has multiple processors, the operating system can schedule each thread to run concurrently on different processors.

Of course, since each thread is its own process, they need to not be blocked by the operating system when another one of them switches into kernel space. Unlike the many-to-one method, this implementation requires some support from the operating system. Recall that threads in a multi-threaded application share the same memory while executing. Therefore, the kernel must provide a means for creating new processes that share memory.

Finally, because each thread is its own process, many of the synchronization techniques (e.g., semaphores and condition variables) carry a large overhead cost due to context switching between user space and kernel space. For example, when waking up a set of threads that are waiting on the condition, a switch between kernel space and user space and back again must occur to wake up each thread. The associated overhead of this operation is two times the number of threads waiting times the cost of a context switch. Not cheap.

Many-To-Many

The many-to-many implementation of threads is nice because it takes the best ideas from both the one-to-one and many-to-one methods while avoiding many of their disadvantages. In the many-to-many model, many kernel processes execute, and each process represents a scheduler that picks a thread to run. Basically, each process acts as its own many-to-one implementation of threads by scheduling different threads to run within that kernel process.

Having multiple processes coordinate to do this solves the problem of all threads being blocked when one thread makes a system call and switches into kernel space. In the many-to-many model, the other kernel processes can still schedule threads to run. Also, the problems with context-switch overhead are mitigated because each process performs the switches between execution of threads. As a result, this switching happens entirely in user space.

Of course, all this functionality comes with a price — this method is the most complicated to implement of the three and requires kernel support for the relationship between kernel threads and user threads. It is the most common implementation on commercial Unix systems (e.g., Solaris, Digital Unix, and IRIX).

The Linux Way

The Linux implementation of *pthreads* follows the one-to-one method. The kernel does not support the functionality required for a many-to-many implementation, and since context-switches in Linux are relatively fast, that disadvantage of the one-to-one method is much less of a problem. Linux also provides the mechanism necessary to create a new process that shares a memory space. You are probably familiar with the `fork()` system call that creates a new process based upon the calling process. Linux provides another system call, `__clone()`, that is more general than `fork()`.

Where `fork()` only gives the child process a copy of the state of the parent process, `clone()` can be used to create a new process that shares *or* copies resources with an existing process. You can share or copy the memory map, file systems, file descriptors, and signal handlers of the existing process. The `fork()` system call is essentially a special case of `clone()` where none of the resources are shared. Let's take a closer look at the `clone()` system call. Its prototype, as found in `<sched.h>`, is listed in *Figure Two*.

Notice that, unlike `fork()`, `clone()` takes in a function, `fn`, to be executed. This is the first argument to the call. When `clone()` successfully completes, `fn` will be executing simultaneously with the calling function.

The `child_stack` argument specifies where the child process should start its stack. Since it is possible for the child and the parent to share a memory map, the parent is responsible for allocating space for the child process's stack. The parent and child can execute different functions after the call to `clone()`, so they cannot share a single stack.

The `flags` argument indicates exactly which parts of the

Table One: Flags for clone()

Flag	Description
`CLONE_VM`	Share the memory between the parent and child process. If this flag is not set, the memory will be copied as is the case with `fork()`
`CLONE_FS`	Share the file system information. A call to a function like `chdir()` will have an effect on both the child and parent process regardless of the caller.
`CLONE_FILES`	Share file descriptors. Any newly created file descriptors after the `clone()` will be valid in both child and parent processes.
`CLONE_SIGHAND`	Share signal handlers. All signal handlers set up by either parent or child after the `clone()` are common to both.
`CLONE_PID`	Share the process ID. The child and parent process will have the same pid. Should not be used with Linux kernel versions after 2.1.97.

that to happen, whether we like it or not. And B) How do you make sure that the technology advances along the principles and the values of the original open source community?

LM: So what comes first for you? What if your board of directors decided that Red Hat had to begin selling proprietary software?

MS: [laughing] What is the upside for me to answer that question? I think it's a personal choice. For me it's not something I would want to participate in. Selling proprietary software puts me in no different position than all kinds of companies selling widgets out there. It's the commitment to open source that differentiates us.

My personal goal was never money. And I think when I came to Red Hat there was a lot of suspicion. I remember sitting down with Mark Ewing and him saying, "you are going to change this into a capitalist organization."

I believe that what's at stake right now is the whole notion of how information will be available for the next generation of citizens, whether they be in Germany or Japan or the United States. In the United States right now, a lot of that policy is being formulated and being discussed without both parties sitting at the table. With [Microsoft VP] Jim Allchin's recent comments to US Senators, the Senate only got one viewpoint. There was no open source representative to challenge his statements and his opinion. There are policy decisions being made on the state and federal levels. If the open source community is not able to establish a broader platform, then we will be forced to live with decisions that are being made.

As an example, Red Hat spends a fairly large sum of money now to battle UCITA (Uniform Computer Information Transactions Act). But that's only one small element. There are many more. And the proprietary vendors are having a relatively easy time with all of their lobbying efforts because the open source community has not been able to galvanize itself to insure that our views are properly represented in Congress.

LM: There is a sense in the open source community that the political process has been irrelevant to the development of important technologies like the Internet. Are you in dialogue with people in the open source community about this?

MS: I have tried to broach this topic with other CEOs in the open source industry and it didn't go too far.

LM: Why not? What did they want to talk about?

MS: Red Hat's dominance of the distribution market, which I think is completely irrelevant.

Robert McMillan is editor at large with Linux Magazine. *He can be reached at bob@linux-mag.com.*

INTERNET ROUTING

By Craig Hunt

They say that any sufficiently advanced technology is indistinguishable from magic, and it's not hard to see why. When you think about the incredible amount of advanced technology that makes it possible to view a Web site or send an e-mail to anyone in the world, it can definitely seem like magic. However, as we all know, there is no magic in the network, just basic engineering.

All networks are composed of computers that are connected to each other via a shared electrical connection (i.e., a wire). A computer can only communicate directly with other computers that share the same physical network. The magic of the Internet is that it makes possible communication between computers on separate physical networks. To do this, a router is needed to interconnect these networks. Routers are computers that contain some intelligence about the topology of the network and have the ability to forward ("route") packets between the networks they are aware of. Without routers, your computer could not communicate with a computer on the other side of town, let alone on the other side of the world.

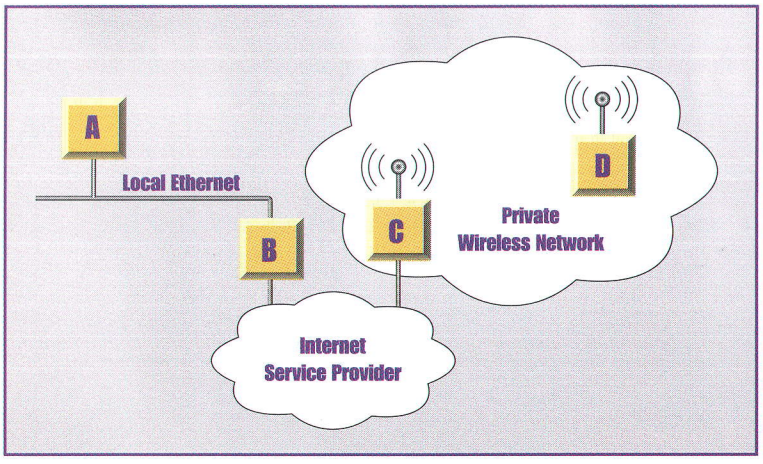


Figure One: Three separate physical networks linked with routers.

Figure One shows three separate networks. Because computers can only communicate directly if they share a network, computer A can talk to B, but A cannot talk to C or D. B can talk to systems A and C, but not to D. C can talk to B and D, but it can't talk to A. D can talk to C, but not to any of the other systems. This is not a global network! A must rely on system B to deliver a message to D. B in turn sends the message to system C, counting on it for delivery. C then sends the message to D. Thus, in *Figure One,* computers B and C are routers. They link together the different physical networks and forward messages between them.

It's worth keeping in mind that even though routers are required for computers to communicate between

Shuttling Packets around the Internet with RIP and RIPv2

networks, routers are not the only machines that make routing decisions. Every network device that uses the Internet Protocol (IP) makes routing decisions. How does this work? Well, in simplified terms, when a Linux system has a packet to deliver, the destination address of the packet is looked up in a routing table. The routing table tells the system whether the packet should be delivered directly to a computer located on this physical network or if it needs to be forwarded to a different physical network.

Listing One: Sample Routing Table

```
# route -n
Kernel IP routing table
Destination   Gateway         Genmask         Flags   Metric   Ref   Use   Iface
172.16.55.0   0.0.0.0         255.255.255.0   U       0        0     0     eth0
172.16.50.0   172.16.55.36    255.255.255.0   UG      0        0     0     eth0
127.0.0.0     0.0.0.0         255.0.0.0       U       0        0     0     lo
0.0.0.0       172.16.55.1     0.0.0.0         UG      0        0     0     eth0
```

Table Based Routing

At this point, it's probably a good idea to take a look at this routing table. While we've included the routing table from one of our systems (running Red Hat 7) in *Listing One,* you can easily view the routing table for your own computer. Just use the *route* command with the *-n* option, and it should come right up. The *-n* option prevents *route* from converting IP addresses to host names, which creates a clearer display.

As you can see in *Listing One,* each entry in the routing table starts with a `Destination` value. The `Genmask` field is the bit mask that will be applied to the destination address of the packet to see if it matches the `Destination` value in the table. For example, the address `172.16.50.183` would match the second entry in this table because performing a logical `AND` of that address with the `Genmask` of `255.255.255.0` yields `172.16.50.0`.

When an address matches an entry in the table, the `Gateway` field tells the system how to reach the specified destination. If the `Gateway` field contains the IP address of a router, then that router is used. If the `Gateway` field contains all zeros (`0.0.0.0`) or an asterisk (`*`), the destination is a directly connected network, and the "gateway" is the

computer's network interface. The last field for each table entry is the network interface used for the route (`Iface`). In this example, it is either the first Ethernet interface (`eth0`) or the loopback interface (`lo`).

Listing Two: Updated Routing Table

```
# route -n
Kernel IP routing table
Destination   Gateway         Genmask         Flags   Metric   Ref   Use   Iface
192.168.0.0   0.0.0.0         255.255.255.0   U       0        0     0     eth0
129.6.0.0     192.168.0.3     255.255.0.0     UG      0        0     0     eth0
127.0.0.0     0.0.0.0         255.0.0.0       U       0        0     0     lo
0.0.0.0       192.168.0.1     0.0.0.0         UG      0        0     0     eth0
```

The remaining four fields display supporting information about the route. The `Ref` field shows the number of times the route has been referenced to establish an outbound connection, and the `Use` field shows the number of packets transmitted via the route. The `Flags` field describes certain characteristics of the route, such as `U`, which means the route is up (i.e., it's "live"), and `G`, which means the route uses an external gateway. The `Metric` field shows the cost of using the route. Routing metrics are used by the routing protocols to determine the "cheapest" way to reach a given destination.

The first line in the sample table defines the connection of this host to the local Ethernet. This entry tells us that the local network is `172.16.55.0` and that the host connects to it through interface `eth0`. The second line tells us that `172.16.55.36` is the gateway to network `172.16.50.0`. The third line defines the loopback network and states that the gateway to it is the interface `lo`. Finally, the last line defines the default route, as indicated by the destination value `0.0.0.0` (Destination contains the keyword "default" on some systems) and the `Genmask` value `0.0.0.0`. If the destination of a packet does not match a specific route, the packet is sent to the default gateway.

On most networks, the only system with a very complex routing table will be the router itself. Host computers tend to have much simpler, static routing tables that only contain three routes:

- The loopback route
- The route to the local network
- The route to the default router

These three routes are statically defined by the system administrator by either answering configuration questions during the initial installation or by using the *route* command to manually add routes to the routing table (which we'll take a look at in a minute). This static table rarely changes once it is created. (Although Linux systems are most often configured as hosts, this does not have to be the case. See the *Creating a Router* sidebar for information about the switch that causes a Linux system to act as a router.)

Adding a Static Route

IP relies on the routing table to direct packets to their destinations, but IP does not build the routing table. Routes enter the routing table in one of two ways — either the system administrator enters them as static routes, or they are added to the table dynamically by a routing protocol. Manually adding static routes works very well when a single default route is used, but building the complex routing tables needed by routers requires a dynamic routing protocol.

The *route* command adds or deletes entries in the routing table. For example, to add the route `129.6.0.0` to the routing table, you would issue the command:

```
# route add -net 129.6.0.0 netmask
   255.255.0.0 gw 192.168.0.3 dev eth0
```

In the sample command, the keyword `add` tells *route* to add a new route to the table. The `-net` value is the destination address of the network reached via this route. The `netmask` value is the address mask that will appear in the `Genmask` field when the routing table is displayed. The `gw` value, which will appear in the `Gateway` field of the routing table display, is the address of the external gateway through which packets are sent to reach this destination address. Finally, the *route* command permits you to specify the network interface to use for the route. Generally, there is no need to define the interface, because the system will choose the correct interface to reach the specified router by default. However, the `dev` option exists so you can define the interface yourself if you want. These four items — the destination, the address mask, the gateway, and the interface — are the basic components of a route. After entering the route, you can issue the command `route -n`; again, you would see the output in *Listing Two*.

The *route* command also accepts an argument called `metric`, which assigns a "cost" to the route. Static routing

CREATING A ROUTER

By default, Linux computers do not forward packets between networks (a primary function of routers). The process of doing this is called IP forwarding. To get a Linux system to act as a router, you must enable IP forwarding by setting */proc/sys/net/ipv4/ip_forward* to 1 as follows:

```
echo "1" /proc/sys/net/ipv4/ip_forward
```

Store this command in a startup file, such as *rc.local*, to ensure that IP forwarding is enabled at every boot.

uses the metric to sort duplicate routes in the routing table. However, duplicate routes have no real use in static routing because the system stops searching the routing table on the first match. This means that even if a given route is down and a router with a slightly higher metric is up and running, static routing will always send packets to the router that is down. Obviously, this is not good. However, it's not a major issue, because dynamic routing makes real use of routing metrics in order to avoid this problem.

Dynamic Routing Protocols

Routing protocols perform two functions — they select the "best" routes and communicate those routes to other routers on the network. Thus, a routing protocol is a recipe for defining and for disseminating routes.

Routing protocols are differentiated by the method they use for determining the best route and by the technique they use for distributing routing information. There are only two major algorithms used to select the "best" route:

> Link state algorithms allow routers to build a database that assigns a cost to every link in the network. A link state protocol selects the route that has the lowest total link cost as the best route. Open Shortest Path First (OSPF) is an example of a link state protocol.

> Distance vector algorithms use a metric that represents the number of routers through which the packet must travel. The fewer routers that must handle the packet, the more preferred the route. Routing Information Protocol (RIP) is an example of a distance vector protocol.

RIP is the routing protocol traditionally used by Unix systems. It is easy to explain and understand, so it is a good place to start a study of routing protocols. Additionally, RIP provides excellent examples of the technical problems and challenges that routing protocols present to software developers and network administrators, and for these reasons, we're going to devote the rest of this article to looking at it.

Routing Information Protocol

RIP is the original routing protocol used on Unix systems and is still included in many Linux distributions. It defines the "best" route as being the route with the lowest routing metric. The RIP routing metric is a number from 1 to 15 that represents the number of routers that traffic must pass through. Each router is referred to as a "hop."

When RIP starts, it broadcasts a request for routing information. Another router running RIP responds to the request by sending an update packet that contains the destination addresses and associated metrics it has in its own routing table. In addition to responding to requests, RIP routers issue updates every 30 seconds. If a router stops issuing updates for 180 seconds, the other routers on the network assume it is dead and delete any routes that go through it. *Figure Two* illustrates the format of a RIP update packet.

It's worth taking a moment to examine the fields in the RIP packet. The `command` field either contains a 1 if this is a request packet or a 2 if it is a response packet. The `version` field always contains a 1 for basic RIP, and the address family field always contains a 2, which indicates that the packet uses IP addresses. The last four word-length fields in *Figure Two* contain a destination address, two words of zeros, and a metric that shows the cost of reaching the destination through the router that sent out the update. These four words are repeated for every destination and metric in the router's routing table.

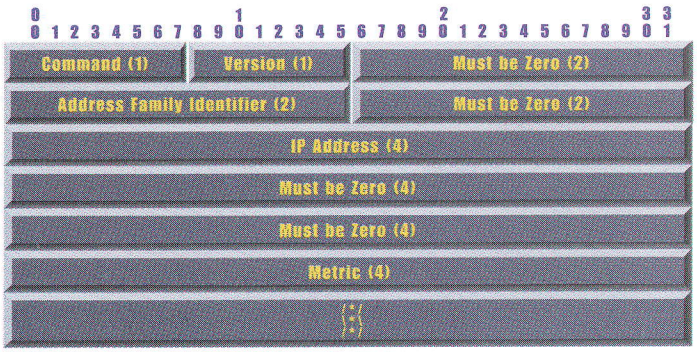


Figure Two: The format of a RIP update packet.

When a router receives an update packet, RIP processes it in the following manner:

> If the update contains new routes that are not in the routing table they are added.

> If the update contains routes that are of lower cost than the same routes in the existing routing table, the new routes are used. The cost of a new route is determined by adding the cost of reaching the router that sent the update to the cost metric included in the update packet.

> If the update contains routes that have a cost of 15, those routes are deleted from the routing table if the update came from the gateway used for those routes. For example, if the routing table contains a route to network `172.16.50.0` through gateway `172.16.55.36`, and it receives a RIP update from `172.16.55.36` with a cost of 15 for the route to `172.16.50.0`, the route is deleted.

Most Unix systems, and many Linux systems, use the *routed* (pronounced "Route 'D'," as in daemon) program to run RIP. When *routed* starts, it issues a request for routing updates and then listens for responses. If it is running on a computer with more than one network interface, *routed* also

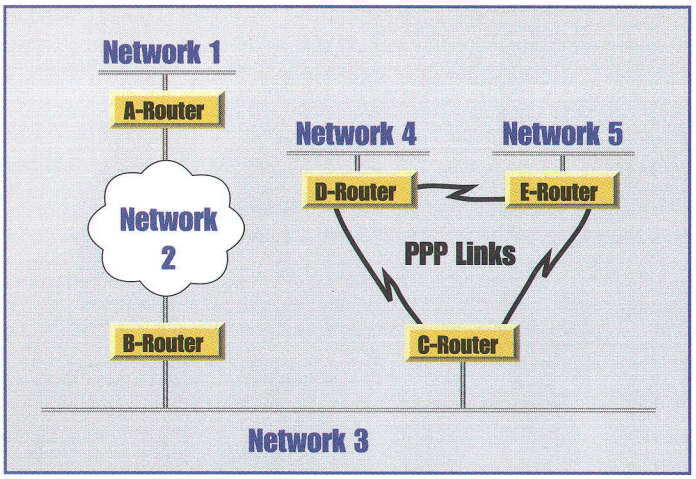


Figure Three: Routers connecting several different networks.

responds to RIP requests with an update packet. Use the `-q` option to prevent *routed* from issuing updates.

routed reads information from */etc/gateways* to build the routing table. The most common use for the */etc/gateways* file is to define a default route, as in this example:

```
net 0.0.0.0 gateway 172.16.12.1 metric 1 active
```

This entry starts with the keyword `net` followed by a network address. (The destination address `0.0.0.0` is used for the default route.) Next is the keyword `gateway` followed by the gateway's address and the keyword `metric` followed by the cost of the route. All */etc/gateways* entries end with either the keyword `passive` or `active`. A `passive` route is a permanent static route placed in the routing table and kept there as long as the system is up. An `active` route can be updated by RIP. `active` routes are used to "prime the pump" during the RIP startup phase, with the expectation that the routes will be updated when the protocol is up and running.

/etc/gateways is the only configuration file for *routed*, and it is rarely needed. On most systems, *routed* learns all it needs to build the correct routing table via the RIP protocol. It does not require manual configuration.

RIP is a simple protocol, and *routed* is a simple program. However, simplicity has a price. RIP has three fundamental limitations:

- Its network diameter is limited to 15 hops.
- It cannot send network bit masks with routing updates.
- It suffers from slow convergence (meaning it can take a long time for the routing table to reflect the current state of the network). *Figure Three* helps illustrate this problem.

Slow Convergence

In order to really understand the problems behind slow convergence and the techniques used to address them, you need to understand a problem known as "count-to-infinity." The problem goes something like this; in *Figure Three*, the c-router reaches network 1 through b-router and then through a-router. Network 1 is two hops away from c- router and one hop away from b-router. Therefore, b-router advertises a cost of 1 for network 1, c-router advertises a cost of 2, and traffic is routed through b-router. This is all fine and dandy until something goes wrong.

Think about a scenario like this; if a-router crashes, b-router will wait for an update from a-router for 180 seconds. While waiting, b-router continues to send updates to c-router that keep the route to network 1 in c-router's routing table. When b-router's timer finally expires, it removes all routes through a-router from its routing table, including the route to network 1. It then receives an update from c-router advertising that c-router is two hops away from network 1. b-router installs this route and announces that it is three hops away from network 1. c-router receives this update, installs the route, and announces that it is four hops away from network 1. This continues until the cost of the route to network 1 reaches 15 in both routing tables. If the update interval is 30 seconds, this could take a long time! RIP deals with the problem of counting to infinity by employing two techniques — split horizon and poison reverse.

Split horizon prevents a router from advertising routes on the link from which those routes were obtained. This solves the problem described above, because c-router does not announce the route to network 1 on network 3 and therefore does not confuse b-router with bad updates. While this feature works for the example described above, it does not work for all count-to-infinity problems (more on this later.).

Poison reverse makes a router advertise routes with an infinite distance on the link from which the routes were obtained. With poison reverse, c-router advertises network 1 with a cost of 15 to all systems on network 3 to positively state that network 1 cannot be reached through c-router.

Split horizon and poison reverse solve the problem described above. But what happens if c-router crashes? Again, refer to *Figure Three*. Because of split horizon, d-router and e-router do not advertise to c-router the route to network 3 because they learned the route from c-router. They do, however, advertise the route to network 3 to each other. When c-router goes down, d-router and e-router perform their own count to infinity before they remove the route to network 3. This problem can be addressed using a technique known as "triggered updates."

With triggered updates, when an external router crashes or a local link goes down, the local router immediately sends an update to its neighbors. This update advertises the routes with infinite cost that were deleted from the local router's routing table and tells the neighbors to also remove them. With triggered updates, neighbors are informed of changes quickly; bandwidth is also used efficiently, because triggered updates include only the routes that have changed.

For an example of how this works, let's look again at *Figure Three*. If c-router crashes, e-router and d-router wait 180

seconds and remove the routes to networks 1, 2, and 3 from their routing tables. They then send each other triggered updates with a metric of 15 for networks 1, 2, and 3. Thus, they tell each other that they cannot reach these networks, and no count to infinity occurs.

Split horizon, poison reverse, and triggered updates handle most slow convergence problems. They do not, however, increase the RIP network diameter or add support for network bit masks. Another major drawback is that RIP cannot disseminate the network bit masks needed to properly interpret routes. This problem alone makes RIP a poor choice for modern networks. Fortunately, an updated version of RIP, known as RIPv2, solves this problem.

RIPv2

RIP Version 2 (RIPv2) adds a "network mask" field and a "next hop address" field to the original RIP packet. Remarkably, RIPv2 adds these capabilities while remaining completely compatible with RIP. RIP and RIPv2 routers can coexist on a single network without any problems. All of the features of RIPv2 are implemented in unused fields of the original RIP packet, as shown in *Figure Four*.

The command, version, address family, IP address, and metric fields are exactly the same as those used in basic RIP, except this time the version field contains a 2. The subnet mask field contains the network bit mask associated with the destination, and the "next hop address" field provides the address of the gateway. In basic RIP, the gateway is always assumed to be the router that sends out the update. The next hop address field specifically identifies the gateway to be used with that packet, allowing a RIPv2 router to provide updates for routers that don't run RIPv2. If the next hop address is `0.0.0.0`, the router that sends the update is assumed to be the gateway for the route.

The two remaining fields, routing domain and route tag, have limited utility. Route tag is not used by RIPv2. The routing domain value is the process number of the RIPv2 process that issued this update if the router runs multiple RIPv2 processes. Routers do not really run multiple RIPv2 processes, so this field always contains 0.

In addition to support for address masks, RIPv2 uses multicasting to reduce the load on systems that do not want

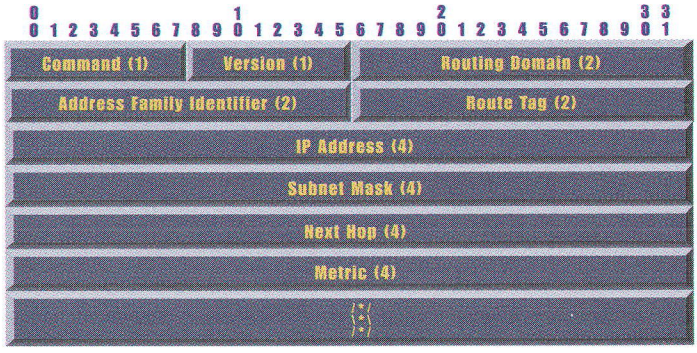


Figure Four: Format for RIPv2 update packet.

FOR MORE INFORMATION

LINKS TO STANDARDS DOCUMENTS:

The RIP protocol standard
ftp://ftp.isi.edu/in-notes/rfc1058.txt

The RIPv2 protocol standard
ftp://ftp.isi.edu/in-notes/rfc2453.txt

The OSPF protocol standard
ftp://ftp.isi.edu/in-notes/rfc2328.txt

GATED LINK:

http://www.gated.org

INTERESTING BOOKS:

Anatomy of an Internet Routing Protocol,
John T. Moy, Addison-Wesley, 1998.

Routing in the Internet,
Christian Huitema, Prentice Hall, 1995.

RIPv2 updates; RIPv2 also provides an authentication scheme to prevent accidental updates from misconfigured hosts. RIPv2 is the only version of RIP that is suitable for use on a modern network.

Whereas *routed* is the program that implements RIP, you would use the *gated* program to run RIPv2. *gated* actually combines several different routing protocols in a single software package. This means that the routing table can be built using multiple protocols and routing policies can be implemented that prefer routes learned from one protocol over those from another. Needless to say, with all that flexibility comes a fair bit of complexity, and for that reason, configuring *gated* is outside the scope of this article. Check out the *gated* man page for more information. Refer to the *For More Information* sidebar for additional sources.

Still Not Perfect

Despite its improvements, RIPv2 is still RIP and still contains some of the same limitations. For example, it uses the same distance-vector algorithm for determining the best route and therefore limits the diameter of the network to 15 hops. The only way to get around this limitation is to use a protocol that employs a different algorithm. The best example of this is the link state protocol OSPF (Open Shortest Path First), which is used for large national networks. If you're interested in learning more about OSPF, please check out the link in the *For More Information* sidebar. Still, if you're just experimenting with routing, or want to convert your Linux box into a simple network router, RIPv2 should be more than adequate for the job.

Craig Hunt is the best-selling author of O'Reilly & Associates' Internetworking with TCP/IP. *He can be reached at craig.hunt@wrotethebook.com.*

GURU GUIDANCE

Administering E-mail

By Æleen Frisch

Making sure that users' electronic mail gets sent out and delivered is one of the system administrator's most important jobs, and it's also one that becomes extremely visible should things go wrong. Inevitably, administering e-mail is time-consuming and frustrating, at least intermittently.

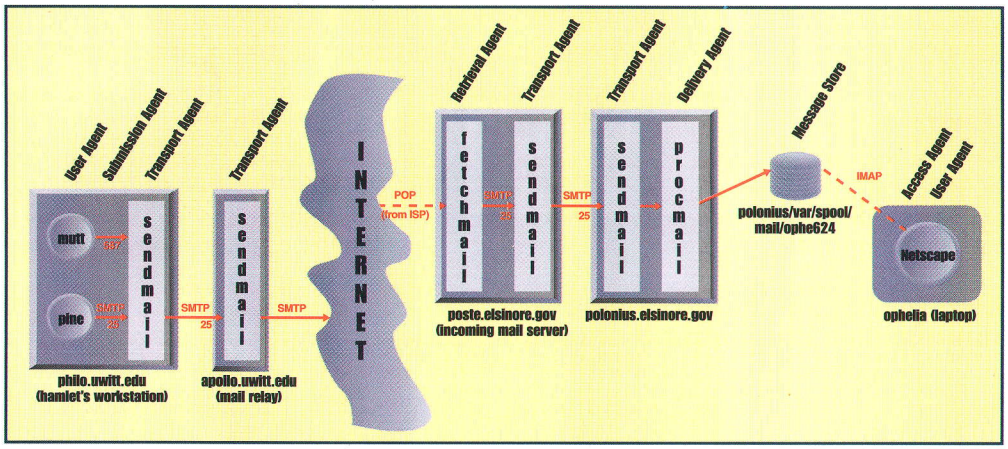


Figure One: How an e-mail from Hamlet makes its way to Ophelia's computer via the Internet.

This column begins a three-part look at some key aspects of the usual Linux electronic mail system. This month, we will begin with an introduction to how electronic mail gets created and look at some of the complexities involved in mail addressing. In future months, we will take a look at the key component responsible for most aspects of mail transportation and discuss how electronic mail may be filtered.

Parts of the Mail System

As with regular postal mail, a fully functioning electronic mail system depends on a series of distinct, and often geographically separated, facilities and processes working together (or at least with mutual cooperation). A mail system is made up of the following types of components:

➤ Programs that allow users to read mail and submit new messages — such programs are known as "user agents." There are a variety of such mail programs available under Linux, ranging from the traditional (and primitive) *mail* command to character-based, menu-driven programs such as *elm*, *mutt*, and *pine* to Internet-integrated graphical packages like Netscape. In addition, some users prefer the mail facilities embedded within their favorite editor (such as *emacs*). In any case, user agents do not typically require a lot of administrative time or attention.

➤ Programs, typically run as daemons, that accept outgoing e-mail ("submission agents"), send it on its way, and begin the process of delivering it to its final destination (system and user). The latter functions are the responsibility of mail "transport agents." *sendmail* is the traditional Unix transport agent and frequently functions as a submission agent as well, although some user agents now incorporate this capability themselves. Current estimates indicate that about 75 percent of all e-mail is transported by *sendmail*. Other transport agents include *Postfix*, *qmail*, and *smail*. Transport agents use the Simple Mail Transfer Protocol (SMTP) to exchange data.

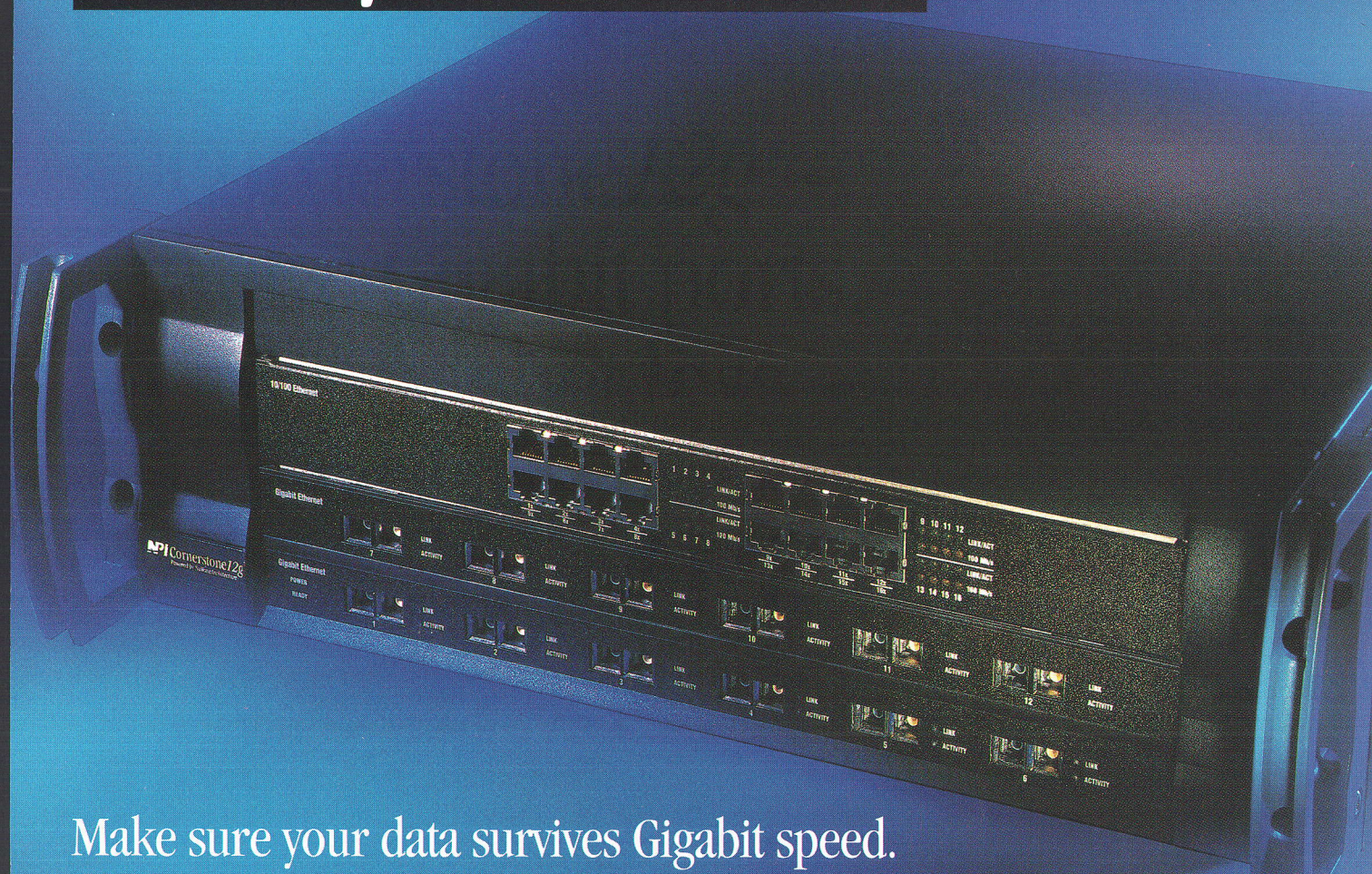
➤ A program that actually places the message into the appropriate user's mailbox. Once mail arrives on its destination system, the transport agent hands it off to a "delivery agent." There may be separate delivery agents for different types of messages (e.g., local vs. remote) and different transport protocols (e.g., SMTP vs. UUCP).

➤ The final destination of a mail message from the point of view of the transport agent may not be the actual location from which the user will access it. There are two situations in which this occurs. The first is when a user, organization, or entire site has only an intermittent connection to the Internet. Messages will be stored for them on their ISP's site until they are ready to collect them. A "retrieval agent" will then periodically establish a connection to the ISP to send new messages and collect those that are waiting. The most common of these programs is *fetchmail*.

The second situation is when a user

A GIANT LEAP AHEAD

Cornerstone*12g* – twelve Gigabit ports, zero packet loss. Only from NPI.



Make sure your data survives Gigabit speed.

With independently verified wire-speed, non-blocking performance, NPI's new Cornerstone *12g* Gigabit Ethernet switch prevents today's fastest applications from jamming your network. That's because NPI switches are "Powered by NuWave*Architecture*," your guarantee of 100% data throughput with zero packet loss at incredibly low prices per port. Contact NPI today at 1-800-674-8855 or visit www.npi.com for OEM and VAR opportunities.

TOLLY Up to Spec CERTIFIED

NuWave ARCHITECTURE

NPI

The Gigabit Ethernet Company™

1-800-674-8855 NETWORK PERIPHERALS INC. www.npi.com

Listing One: struct Defining a Thread

```
struct _pthread_descr_struct {
    pthread_descr p_nextlive, p_prevlive;
                                  /* Double chaining of active threads */
    pthread_descr p_nextwaiting;  /* Next element in the queue holding the threads */
    pthread_descr p_nextlock;     /* can be on a queue and waiting on a lock */
    pthread_t p_tid;              /* Thread identifier */
    int p_pid;                    /* PID of Unix process */
    int p_priority;               /* Thread priority (== 0 if not realtime) */
    struct _pthread_fastlock * p_lock; /* Spinlock for synchronized accesses */
    ...
    /* New elements must be added at the end.  */
} __attribute__ ((aligned(32))); /* We need to align the structure so that
                                    doubles are aligned properly. This is 8
                                    bytes on MIPS and 16 bytes on MIPS64.
                                    32 bytes might give better cache
                                    utilization.  */
```

parent should be shared with the child. *Table One* (pg. 66) shows a list of the flags that may be bitwise-or'ed together and passed in the call to `clone()`. Besides the sharing flags, the lowest byte of the `flags` argument represents the number of the signal to be sent to the parent when the child process dies.

The last argument to `clone()`, `arg`, represents the argument that is to be passed to the function, `fn`, that `clone()` will execute. The `__clone()` function returns the process ID of the child process when it succeeds. It will return -1 when it fails. See the man page for more information about the failure cases.

If you've been following this column's series of articles on threads over the past few months, you may be thinking that the functionality of `clone()` seems awfully familiar... Well, that's because it is pretty much how the `pthread_create()` function behaves, and the similarity is no coincidence. As stated in the man page for `clone()`, "the main use of `__clone()` is to implement threads." The Linux kernel developers felt that the one-to-one method of implementing threads was the way to go, and therefore designed the kernel to enable threads support in that way. The implementors of the *pthreads* library simply followed their lead.

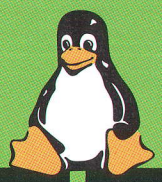
Listing Two: The pthread_create() Function in pthread.c

```
/* Thread creation */

int __pthread_create_2_1(pthread_t *thread, const pthread_attr_t *attr,
                         void * (*start_routine)(void *), void *arg)
{
    pthread_descr self = thread_self();
    struct pthread_request request;
    if (__pthread_manager_request < 0) {
        if (__pthread_initialize_manager() < 0) return EAGAIN;
    }
    request.req_thread = self;
    request.req_kind = REQ_CREATE;
    request.req_args.create.attr = attr;
    request.req_args.create.fn = start_routine;
    request.req_args.create.arg = arg;
    sigprocmask(SIG_SETMASK, (const sigset_t *) NULL,
                &request.req_args.create.mask);
    __libc_write(__pthread_manager_request, (char *) &request, sizeof(request));
    suspend(self);
    if (THREAD_GETMEM(self, p_retcode) == 0)
        *thread = (pthread_t) THREAD_GETMEM(self, p_retval);
    return THREAD_GETMEM(self, p_retcode);
}
```

The Structure of Threads

Now that we understand the `clone()` system call, let's return to our discussion about the implementation of threads under Linux. In the following, we will be referring to the implementation of *pthreads* found in *glibc* version 2.1.3. All of the files mentioned are in the *linuxthreads* directory unless otherwise specified.



WELCOME TO THE REVOLUTION

LINUX
MAGAZINE
THE CHRONICLE OF THE REVOLUTION

www.linux-mag.com

The basic structure of the implementation of threads is as follows: when the first thread is created (i.e., in the first call to `pthread_create()`), a manager thread is created to manage the subsequent creation and termination of threads. The manager will perform the actual creation of new threads. Once the manager is created and initialized, the thread that wishes to create a new thread sends a message to the manager via a pipe and then suspends itself until the manager satisfies the request and wakes it up.

The thread manager spaces each of the thread's stacks 2 MB apart. At the top of each thread's stack resides a `struct` containing the relevant information about that thread (e.g., it's process ID, etc.) for use by the manager and other threads. The definition of that `struct` is found in `internals.h`, and the first few fields can be seen in *Listing One* (pg. 68) (there are more than 40 fields in this `struct`).

To get a better idea for how threads are laid out in memory, let's look at a picture of the stack after a few threads are created (see *Figure One,* pg. 64). Since stacks grow down on most architectures, we draw our picture assuming that this is the case in our example. Notice that the initial thread lies at the top of the stack. This is the thread that represents the `main()` function in an application. The thread descriptor for this thread is kept in the space for global variables and is defined in `pthread.c`. Below the initial thread on the stack is the thread manager's stack (it's descriptor is also in global space and defined in `pthread.c`), followed by the `struct` for each thread and its stack. The manager keeps track of each thread by keeping track of the top of the thread's stacks and, therefore, a reference to their respective `structs`.

Creating New Threads

Now that we have a clear picture of the memory layout, we can look at the code that does the work of creating new threads (see *Listing Two,* pg. 68). When a program calls `pthread_create()`, the `__pthread_create_2_1()` function is called in the *glibc* library — the file called `pthread.c`. First, the function gets the reference to itself (`thread_self()` returns a pointer to the `struct_pthread_descr_struct`).

Next, the function creates a `request` to be sent to the thread manager. Notice that the request is of type `REQ_CREATE`. The `request` also contains the function to be run (`start_routine`), its argument (`arg`), and the attributes (`attr`) the new thread should have. After indicating that the current thread should not be disturbed by any signals through the call to `sigprocmask()`, it writes the create request to the write end of the pipe to communicate that information with the manager. Then the thread suspends itself, waiting for the manager to

Chicks dig Linux!

Agenda™ COMPUTING

Linux is sexy...ok, Linux is cool.
Linux will make you more attractive...
well, not really.

But Linux does run the Agenda VR3, the world's first ever Pure-Linux™ handheld computer. We've built it from the ground up on the embedded Linux VR operating system with an impressive 32-bit processor blazing along at 66MHz, 8MB RAM and a whopping 16MB of Flash Memory.

The VR3 boasts super-high contrast, high-resolution screens that are backlit for use in low-light, Infrared and IrDA data transfer, audio capability and will sync to all major PIMs. It comes in three really cool colors; Shark, Matrix and H_2O (the first transparent PDA), includes a great case to protect your PDA, and a smooth finish that feels yummy to your hand.

PURE LINUX™ "The Only Pure-Linux PDA"

Agenda VR3 The Ultimate Mindtool™

www.agendacomputing.com

wake it up once the request has been completed. If the manager successfully created the new thread, the argument passed into `pthread_create()` is set, and the return code is returned.

The code for the thread manager is found in the file `manager.c`. The main function of this file is `__pthread_manager()`, which sits in a `while()` loop, polling the read end of the communication pipe. Whenever it sees that there is information to be read, it reads the request, calls the appropriate handler, and continues to poll. The snippet of code that handles the REQ_CREATE request is shown in *Listing Three*. Note that the function that does the creation is `pthread_handle_create()`, which is also in "manager.c" and is shown in *Listing Four*.

Listing Three: Code That Handles the REQ_CREATE Request

```
/* The server thread managing requests for thread creation and termination */

int __pthread_manager(void *arg)
{
    ... /* Initialization and signal handling omitted for brevity. */
    /* Enter server loop */
    while(1) {
        n = __poll(&ufd, 1, 2000);
        ... /* Other checks omitted for brevity. */

        /* Read and execute request */
        if (n == 1 && (ufd.revents & POLLIN)) {
            n = __libc_read(reqfd, (char *)&request, sizeof(request));
            ASSERT(n == sizeof(request));
            switch(request.req_kind) {
            case REQ_CREATE:
                request.req_thread->p_retcode =
                    pthread_handle_create((pthread_t *) &request.req_thread->p_retval,
                                          request.req_args.create.attr,
                                          request.req_args.create.fn,
                                          request.req_args.create.arg,
                                          &request.req_args.create.mask,
                                          request.req_thread->p_pid,
                                      request.req_thread->p_report_events,
                                      &request.req_thread->p_eventbuf.eventmask);
                restart(request.req_thread);
                break;
            /* Other cases */
                ...
            }
        }
    }
}
```

Listing Four: The pthread_handle_create() Function

```
static int pthread_handle_create(pthread_t *thread, const pthread_attr_t *attr,
                                 void * (*start_routine)(void *), void *arg,
                                 sigset_t * mask, int father_pid,
                                 int report_events,
                                 td_thr_events_t *event_maskp)
{
   /* Initialization of local variables. */
   ...

   /* Find a free segment for the thread, and allocate a stack if needed */
   for (sseg = 2; ; sseg++)
      {
        if (sseg >= PTHREAD_THREADS_MAX)
         return EAGAIN;
        if (__pthread_handles[sseg].h_descr != NULL)
         continue;
        if (pthread_allocate_stack(attr, thread_segment(sseg), pagesize,
                                   &new_thread, &new_thread_bottom,
                                   &guardaddr, &guardsize) == 0)
         break;
      }
   __pthread_handles_num++;
   /* Allocate new thread identifier */
   ...
   /* Initialize the thread descriptor.  Elements which have to be
      initialized to zero already have this value.  */
   ...
   /* Initialize the thread handle */
   ...
   /* Determine scheduling parameters for the thread */
   ...
   /* Finish setting up arguments to pthread_start_thread */
   new_thread->p_start_args.start_routine = start_routine;
   new_thread->p_start_args.arg = arg;
   new_thread->p_start_args.mask = *mask;
   /* Raise priority of thread manager if needed */
   ...
   /* Do the cloning.  We have to use two different functions depending
     on whether we are debugging or not.  */
   pid = 0;      /* Note that the thread never can have PID zero.  */
   ...
   if (pid == 0)
      pid = __clone(pthread_start_thread, (void **) new_thread,
                    CLONE_VM | CLONE_FS | CLONE_FILES | CLONE_SIGHAND |
                    __pthread_sig_cancel, new_thread);
   /* Check if cloning succeeded */
   if (pid == -1) {
      ...
   }
   /* Insert new thread in doubly linked list of active threads */
   new_thread->p_prevlive = __pthread_main_thread;
   new_thread->p_nextlive = __pthread_main_thread->p_nextlive;
   __pthread_main_thread->p_nextlive->p_prevlive = new_thread;
   __pthread_main_thread->p_nextlive = new_thread;
   /* Set pid field of the new thread, in case we get there before the
      child starts. */
   new_thread->p_pid = pid;
   /* We're all set */
   *thread = new_thread_id;
   return 0;
}
```

Much of `pthread_handle_create()` has been omitted for space purposes. For parts that are left out, the comments have been kept so you can see what occurs there. The code that remains is the guts of the thread creation mechanism. First, this function allocates a new stack for the new thread through the `pthread_allocate_stack()` function. It then sets the necessary fields in the thread descriptor (including the start function and the argument to it). Next, it calls the `__clone()` function with the flags discussed above to share all attributes with the child. The `pthread_start_thread()` function that is run from `clone()` essentially performs some more initialization of the thread and finishes by calling the `start_routine()` function. And thus, the new thread is born.

There's Always More...

Hopefully, you have enjoyed this whirlwind tour of the implementation of threads under Linux. If you would like to learn more about the `clone()` system call and how it relates to `fork()`, take a look at the book *Linux Kernel Internals*, by Beck, Hohme, Dziadzka, Kunitz, Magnus, and Verworner. As you might imagine, the implementations of the two functions are very similar to each other.

If you are interested in learning more about the implementation of threads (including the other pthreads functions), the code in *glibc/linuxthreads* is very well documented and is quite easy to read. Also, that directory contains a *README* and a *FAQ.html* that provide even more information about the use of threads. In the meantime, happy hacking!

Benjamin Chelf is a freelance author and engineer at CodeSourcery, LLC. He can be reached at chelf@codesourcery.com.

PERL OF WISDOM

Processing Footnotes

By Randal L. Schwartz

I wrote a Web page the other day and realized that I wanted footnotes. I wanted to keep the main message in the main text and have annotations for some of the side points. It's easy enough to do, right? Just put some text in a table at the end, use those cute little `sup` tags around the footnote numbers, and hack away.

Oops...those little numbers! I started to dread getting six footnotes inserted and then having to go back to insert yet another between numbers 2 and 3. It was going to be a maintenance nightmare. Could Perl help? Of course!

Expending about 10 times the amount of labor I would have spent doing this manually, I hacked out the program in *Listing One*. This obviously wasn't very efficient, so to make the time that I invested worthwhile, I'll pass the program along to you. Besides, it illustrates how to create an angly-bracket metalanguage for your HTML and XML processing. Yeah, that justifies it.

I started to dread getting six footnotes inserted and then having to insert yet another. It was going to be a maintenance nightmare.

The idea is to insert a footnote into the main flow using a made-up tag of `foot`. The processor pass then takes those out, replacing them with an anchor link and a unique number. Then, at the end of the file, all the footnotes are dumped out. For an example, look at the end of the program. And, I couldn't stop there, so I decided to allow nested footnotes (like those found on the `alt.sysadmin.recovery` newsgroup). About half of my coding time was spent getting those to work. Someday, I will learn to prioritize.

Let's see what I wasted an hour on, starting with the first few lines that begin nearly every program I write. These lines enable warnings, turn on the normal compiler restrictions for non-trivial programs, and disable buffering on `STDOUT`.

Line 5 pulls in the `HTML::Parser`, a wonderful piece of work maintained by Gisle Aas. This is a C-based module for lightning-fast parsing of anglybracket data input, which is normally HTML text. This is much faster than hand-rolled regular expressions. You'll find this as part of the LWP module family in the CPAN.

Lines 7 and 8 contain the footnote list and footnote stack index, respectively. The first item of `@feet` is the text of the first footnote, numbered `1`. The *last* item of `@feet_index` (if any) is the subscript of `@feet` of the current footnote we are creating. As each new `foot` tag is seen, we create a new empty footnote in `@feet` and put its index at the end of `@feet_index`. When the note is ended, we pop off `@feet_index`, thus resuming the previous note. If there are no items in `@feet_index`, it's the main body, and we can just copy the data through.

Yes, this is the logic that it took me the better part of an hour to get cleanly. I wanted the footnotes to be numbered in the order of the start tags. I kept creating algorithms to number them on the basis of the end tag instead — until I formulated the indirection table.

Line 10 keeps track of our nesting of elements. With the way I'm using `HTML::Parser`, it wouldn't matter to the parser that I have mismatched tags. However, since my footnote processing is fragile under these circumstances, I enforced the XML-notion of "well-formedness" and required properly balanced tags.

You may ask why I did not just use `XML::Parser` instead of `HTML::Parser`. Well, I like the callback flexibility of `HTML::Parser` for small projects.

Speaking of which, lines 12 through 17 define my parser object. I defined three callbacks. The first one is for text items, which will get the text as its first parameter. The second one is for start tags, which will get both the original text and the tagname extracted. Finally, the end tags are also triggered, again getting both the original text and the tagname.

Line 19 turns on `xml_mode` in the `HTML::Parser`, which keeps the tags in their original case and slightly alters the handling of a few other constructs. Again, this is yet more evidence that I really wanted an XML Parser.

Line 20 pulls in the contents of the `DATA` filehandle, which is the contents of this file under the `__END__` token below, with my sample data (the description of how I write my Perl columns). The result of this parsing pass is a number of calls to the three callback subroutines, which after completion will have printed the main part of the text to `STDOUT` already. We'll see how this works shortly.

RackSaver™

YOU HAVE THE POWER

AMD

AMD gives you the power!
RackSaver gives you the space!
With the introduction of our
UltraThin line of servers
RackSaver brings the Athlon
to the ISP and Beowulf cluster market
in the smallest chassis possible.
Athlons with DDR NOW!!
Dual processor in 'Q1 2001 for
unmatched performance. Go to our
website for instant configuration
and pricing of our systems.

UltraThin 1100 Server

- World's First AMD 1U Server
- Over 1 GFLOP!
- 1GHz CPUs available
- Starting at just $999.00

*Thin*Server 2100

- Huge Storage & Redundant Power
- Over 500GB per system in 2U
- Hot-Swap drive bays available
- Hot-Swap power supplies also

*Thin*Server 2300

- 3 Systems in 2U of Space
- Up to 63 Systems in One Rack
- Over 100 GFLOPs in One Rack
- Cheapest $$$ per GFLOP

Now Introducing!

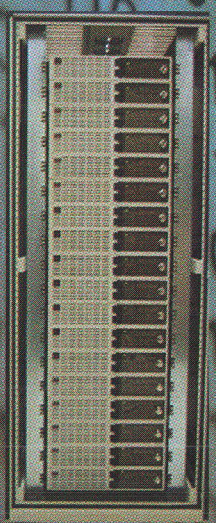
UltraThin 1200 Server

- 2 Systems in 1U of Space!
- Up to 84 Systems in a Single Rack
- 2 drives and 1 cdrom per system
- Ultimate redundant solution for ISP
- Massive power for clusters

Custom Chassis Available!

Linux POWERED

Theirs



18 nodes

OURS

84 NODES!!!

AMD

Athlon™ PROCESSOR

RackSaver Inc.
7100 Convoy Court
San Diego, CA 92111
Ph#(858) 874-3800
Fax(858) 874-3838

WWW.RACKSAVER.COM

The other effect of the parse is the extraction of the footnotes into `@feet`. Lines 22 to 27 dump this out in a nice way. I'm using an HTML table for layout, with a column of the footnote numbers and a column of the footnote text. Each footnote also has an anchor assigned to it, which we can use as the target of an internal anchor reference using a fragment identifier. Lacking inspiration, I numbered these `note1`, `note2`, and so on.

Line 29 is an exit, redundant because there's only subroutines from here down; it exists just to keep it clear where the program ends.

Lines 31 to 38 define the text handler, which is called whenever `HTML::Parser` finds some text outside a tag. I've selected to pass the text itself as the first parameter, which I copy into `$text` in line 32. If we're currently in a footnote, this text is part of the footnote, so we append it to the right footnote. However, the right footnote will be the one whose index is in the rightmost element of `@feet_index`, hence the indirection. Remember, this is not necessarily the highest indexed footnote (I might have nested footnotes, but this would be the exception and not the rule). If I'm not in a footnote, the text is simply dumped to `STDOUT` instead.

Lines 40 to 51 define the handler for the beginning of every element. The incoming parameters are the original text and the tagname for that start tag (we don't need the attributes). These are assigned to `$text` and `$tagname` in line 41.

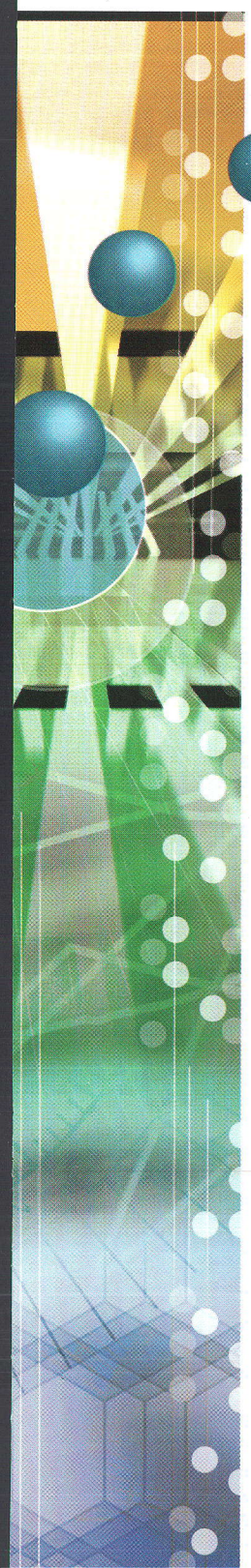
Line 44 notes the current element name by pushing the tag onto the stack. I'll check this on the close tag in order to make sure that the tags are nested properly to make valid elements.

Line 46 does the work for a footnote start tag. First, I created the new footnote as empty in line 47. Next, I inserted the reference to the footnote in line 48 by faking a text event containing the reference. I can't just print this because I

Listing One: Using Footnotes in HTML — Part I

```
1  #!/usr/bin/perl -w
2  use strict;
3  $|++;
4
5  use HTML::Parser;
6
7  my @feet;              # final footnote list
8  my @feet_index;        # indexes into @feet
9
10 my @elements;   # ensure nested tags match
11
12 my $parser = HTML::Parser->new
13    (
14     text_h => [\&text_h, "text"],
15     start_h => [\&start_h, "text, tagname"],
16     end_h => [\&end_h, "text, tagname"],
17    );
18
19 $parser->xml_mode(1);    # keep tags
                               case-sensitive
20 $parser->parse_file(\*DATA);
      # prints main part to STDOUT
21
22 if (@feet) {           # we had footnotes?
23     print "<hr><table border='0'
          cellspacing='0' cellpadding='2'>\n";
24     print "<tr><td valign='top'><sup><a
          name='note$_'>$_</a></sup></td>",
25         "<td>$feet[$_-1]</td></tr>\n"
             for 1..@feet;
26     print "</table>";
27 }
28
29 exit 0;                    # end of code
30
31 sub text_h {
32     my ($text) = @_;
33     if (@feet_index) { # are
                        we inside a footnote?
34         $feet[$feet_index[-1]] .= $text;
              # append to that
35     } else {
36         print $text;          # just show it
37     }
38 }
39
40 sub start_h {
41     my ($text, $tagname) = @_;
42
43     ## ensure proper nesting
44     push @elements, $tagname;
45
46     if ($tagname eq "foot") {
47         push @feet, "";    # the note itself
48         text_h("<sup><a href='#note".@feet.
                "'>".@feet."</a></sup>");
49         push @feet_index, $#feet;    #
              pointer to note
50         return;
51     }
52
53     text_h($text);    # uninteresting start tag
54 }
55
56 sub end_h {
57     my ($text, $tagname) = @_;
58
59     ## ensure proper nesting
```

LinuxWorld™ CONFERENCE & EXPO



Where Open Source Comes of Age

Don't miss the world's leading Linux event focusing on Linux and Open Source solutions. Whether you're looking for a better way to manage and scale your network, or a more cost-effective way to run your company, LinuxWorld Conference & Expo can show you how. Be there to network, learn and see where Open Source comes of age.

- Hear **Keynotes** from leading Linux and open source visionaries
  - **Shane Robison**, Sr. Vice President & CTO, Compaq Computer Corporation
  - **Larry Lessig**, Law Professor, Stanford Law School
- Attend Feature Presentations such as **"Golden Penguin Bowl"** hosted by Larry Augustin of VA Linux Systems and **"The State of Open Source"** featuring Linus Torvalds
- Over 250 **leading hardware and software vendors** showcasing never before seen Linux applications, hardware, services and embedded systems
- The first annual **Developer's Conference** geared exclusively to application developers, along with the most comprehensive and very best Linux educational training available in the world
- The **Embedded Linux Pavilion** in partnership with the **Embedded Linux Consortium**

Conference
August 26-30, 2001

Expo
August 28-30, 2001

Moscone Convention Center
San Francisco, CA

Owned & Managed By:
IDG WORLD EXPO

Cornerstone Sponsor	Platinum Sponsors	Media Sponsors
VA LINUX SYSTEMS	COMPAQ, GreatBridge, hp invent, IBM, intel, SERVE LINUX, sgi, Sun microsystems	CLiCK, COMPU-MART, COMPUTERWORLD, CMP Dr.Dobb's, EC TECHNOLOGY NEWS, LINUX JOURNAL, LINUX, NetworkWorld, SD Times, Server/Workstation EXPERT, Software BUSINESS, SOFTWARE MAGAZINE, Sys Admin

www.linuxworldexpo.com

might still be in another footnote; faking the text event "does the right thing." (For better maintenance, I'd probably pull the "add text" operation to a separate subroutine that both the text handler and this handler call, but this worked for this quick-and-dirty program.)

Line 49 adds the footnote index onto the footnote stack. Note that we cannot do this before the previous line because the footnote reference would end up inside itself.

Line 53 handles the start tags that are of no interest (everything except `foot` tags) by simply copying them as-is to the current output (either a footnote or `STDOUT`).

Lines 56 to 72 handle the end tags. Again, the text and tagname end up in variables, defined in line 57.

Lines 60 through 64 handle the verification of properly nested tags. If there's no start tag, or the tags don't match, a swift and painless death is the result.

Lines 66 to 69 handle the `foot` end tag, which is the only one of interest. If I find one, I simply pop an entry off the `@feet_index` array, which will pop us back to the previous footnote on the next text item seen. If we don't find any, we're back to dumping to `STDOUT`.

Line 71 dumps the other uninteresting end tags as needed.

That's all there is. It's not rocket science, but it gets the job done. As sample text, I included an outline of what it takes to write a Perl column. If you run the program, you will get the HTML output shown in *Listing Two*.

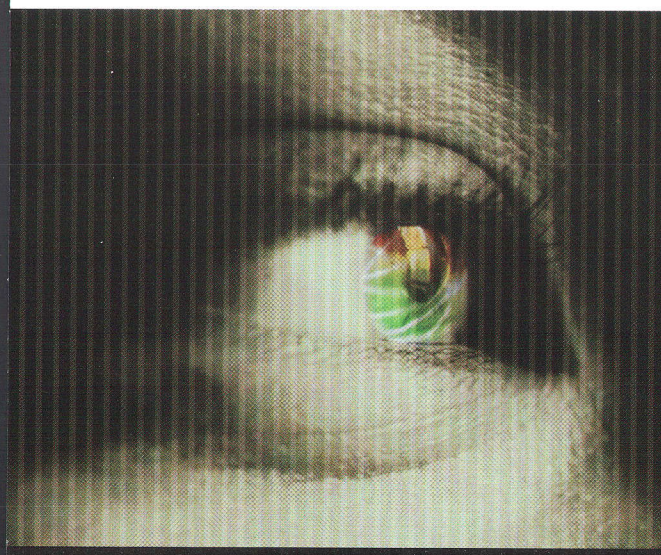
Note how the footnotes have been replaced with internal fragment references and the content of the footnotes has become a table at the end. Yes, I could have done all this by hand, but it was more fun to write the program and get it done right, once and for all. So, don't fear footnotes and writing tiny metalanguages for those odd tasks.

Randal L. Schwartz is the chief Perl guru at Stonehenge Consulting and co-author of Learning Perl *and* Programming Perl. *He can be reached at merlyn@stonehenge.com. Code listings for this column can be found at:http://www.stonehenge.com/merlyn/LinuxMag/.*

Listing One: Using Footnotes in HTML — Part II

```
60      die "saw $text outside of element"
61          unless @elements;
62      die "saw $text nested inside
            <$elements[-1]>"
63          unless $elements[-1] eq $tagname;
64      pop @elements;
65
66      if ($tagname eq "foot") {
67          pop @feet_index; # no longer
                             accumulating here
68          return;
69      }
70
71      text_h($text);  # uninteresting end tag
72  }
73
74  __END__
75  <h2>Writing a Perl column</h2>
76  Writing a magazine column about Perl is
        a simple<foot>for some!</foot> task.
77  Just perform the following steps:
78  <ol>
79  <li>Think of a problem to
80      solve<foot>You can ask around for
            help here.
81      I keep an archive of "todo" ideas,
            and it really helps.</foot>.</li>
82  <li>Write the code to solve it.<foot>The
        code should be between 50 and 200
83      lines for optimum column
84      length<foot>About 10,000 characters.
            </foot>.</foot></li>
85  <li>Fret over the code for a few hours
            <foot>Or a few minutes.</foot>.</li>
86  <li>Write the column<foot>I use POD<foot>See
            <tt>perldoc perlpod</tt>.</foot>
87      format.</foot>.</li>
88  <li>Show the column to a group of friends
89      on IRC<foot>Usually the
                <tt>#perl</tt> channel.</foot>
90      for a quick peer review.</li>
91  <li>Turn it in<foot>By email.</foot>
        to the editor.</li>
92  <li>Wait a few days for the
        galleys<foot>Usually a
93      PDF<foot><i>Portable Document
                Format</i> from
94      Adobe<foot>See <tt>www.adobe.com</tt> for
                downloads.</foot>.</foot>.</foot>
95      to come back.</li>
96  <li>Grimace over the hacks to your lovely
97  prose<foot>just kidding, guys!</foot>
        and provide corrections to
98  the corrections.
99  </li>
100 <li>Wait a few months<foot>Or so it
        seems, since the deadline
101 for an April cover is usually the first
        week of January.</foot> for it
102 to "hit the stands".</li>
103 <li>Wave the magazine in front of your
        friends<foot>Or the cute girl
104 at the bookstore checkout
        counter.</foot>!</li>
105 </ol>
```

The Best Internet Conference in the World Is Now The Best Deal in the Midwest!



THE FOURTH ANNUAL
Internet World Chicago 2001
Conference: July 9–12
Exhibits: July 10–12
McCormick Place
Chicago

Act Now to Save BIG on the BEST Internet Conference in the Midwest!

- **EXPLORE** every available option for increasing your company's productivity and profitability on the Internet.
- **FIND** out what's working, what's tanking and what lies ahead for e-business
- **HEAR** success (and horror) stories from the Internet's top performers and pioneers
- **LEARN** how thousands of other Midwest companies are leveraging the Internet to improve their business
- **COME** away with practical solutions and concrete do's and don'ts you can implement immediately to boost your e-business performance

Produced by:
Penton

Official Sponsor:
INTERNET WORLD

Media Sponsors:
INTERNET WORLD.com
boardwatch
internet.com

FOR MORE INFORMATION and online registration visit www.internetworld.com

To exhibit e-mail eventsales@iw.com

INTERNET WORLD
Chicago 2001

Code: N45

Listing Two: The HTML Output of Listing One

```
<h2>Writing a Perl column</h2>
Writing a magazine column about Perl is a simple<sup>[1]</sup> task.
Just perform the following steps:
<ol>
<li>Think of a problem to
  solve<sup>[2]</sup>.</li>
<li>Write the code to solve it.<sup>[3]</sup></li>
<li>Fret over the code for a few hours<sup>[5]</sup>.</li>
<li>Write the column.</li>
<li>Show the column to a group of friends
  on IRC<sup>[6]</sup>
  for a quick peer review.</li>
<li>Turn it in<sup>[7]</sup> to the editor.</li>
<li>Wait a few days for the galleys<sup>[8]</sup>
  to come back.</li>
<li>Grimace over the hacks to your lovely
prose<sup>[9]</sup> and provide corrections to
the corrections.
</li>
<li>Wait a few months<sup>[10]</sup> for it
to "hit the stands".</li>
<li>Wave the magazine in front of your friends<sup>[11]</sup>!</li>
</ol>
<hr><table border='0' cellspacing='0' cellpadding='2'>
<tr><td><sup>[1]</sup></td><td>for some!</td></tr>
<tr><td><sup>[2]</sup></td><td>You can ask around for help here.</td></tr>
<tr><td><sup>[3]</sup></td><td>The code should be between 50 and 200
  lines for optimum column
  length<sup>[4]</sup>.</td></tr>
<tr><td><sup>[4]</sup></td><td>About 10,000 characters.</td></tr>
<tr><td><sup>[5]</sup></td><td>or a few minutes</td></tr>
<tr><td><sup>[6]</sup></td><td>Usually the <tt>#perl</tt> channel</td></tr>
<tr><td><sup>[7]</sup></td><td>By email.</td></tr>
<tr><td><sup>[8]</sup></td><td>usually a PDF.</td></tr>
<tr><td><sup>[9]</sup></td><td>just kidding, guys!</td></tr>
<tr><td><sup>[10]</sup></td><td>Or so it seems, since the deadline
for an April cover is usually the first week of January.</td></tr>
<tr><td><sup>[11]</sup></td><td>Or the cute girl
at the bookstore checkout counter.</td></tr>
</table>
```

2001 USENIX Annual Technical CONFERENCE

THE HOTTEST ANNUAL TECH CONFERENCE TO DATE

DON'T MISS OUT!

THIRTY TUTORIALS in all, seventeen brand-new. Here's just a sampling:

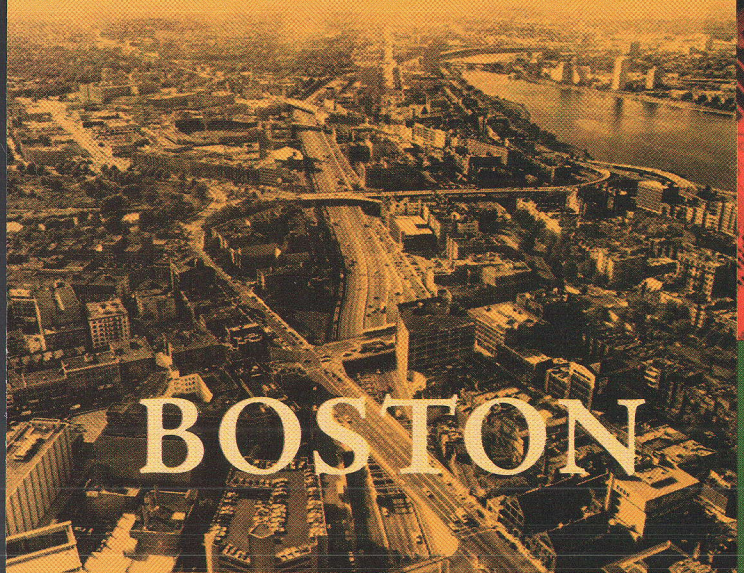
- Blueprints for High Availability
- Examine the Linux Kernel
- Master Advanced Solaris Administration
- Explore the Potential of LDAP

Keynote Speech by Daniel D. Frye, Director of IBM Linux Technology Center

Check out the FREENIX Track

FREE VENDOR EXHIBITION

- LDAP
- Linux
- Large Network Design
- High Availability
- Perl
- Cryptography
- Network Forensics
- Wireless Networking



June 25–30, 2001

TECH SESSIONS: Thursday–Saturday

EXHIBITION DATES: June 28–29, 2001

Boston, Massachusetts

Sponsored by

USENIX

The Advanced Computing Systems Association

http://www.usenix.org/events/usenix01

TECH SUPPORT

Top Tech Support Questions

By Drew Streib

I want to get a bandwidth usage graph for my machine, but don't have a switch with SNMP. Can I get this directly from the linux box?

You can get a lot of information from a Linux machine via SNMP, including network traffic to build bandwidth graphs. I'll explain a method using *mrtg* (Multi Router Traffic Grapher).

You'll first need to install the SNMP daemon, preferably using your distribution's package management system. The installation should start the SNMP daemon. You can check for this by running the following:

```
# ps aux | grep snmpd
```

You should see an *snmpd* program running here. If you do not, you may need to start the daemon yourself. Depending on the distribution you are running, you'll probably need to type:

```
# /etc/init.d/SNMPd start
```

or

```
# /etc/rc.d/init.d/snmpd start
```

If this starts successfully, you'll want to make sure that *snmpd* is appropriately started by *init* in your startup scripts. If this does not start the SNMP daemon, or if an *init* file does not exist, you should check to see if *SNMPd* is, in fact, installed.

There is a *lot* to learn about SNMP. For now, you need to know that it provides several ways to access and control data, and it organizes much of this information in groups called "communities." By default, most Linux installations will create a community called *public* that will contain basic information. This should be available on the local machine via *localhost*.

You can test this by using a program called *snmpwalk* to peruse the exports on your machine. To do this, try issuing the command:

```
# snmpwalk -c public localhost
```

You should see a lot of information dumped to screen. Anything less than this and you'll need to ensure the operation of *snmpd* and that a *public* community has been created and is viewable by *localhost*. Since there are a myriad of options here, if this default setup doesn't work for you, please visit the SNMP page at http://net-snmp.sourceforge.net/.

Once *snmpd* is running on your system and access is properly set up, you will be able to use programs like *mrtg*, which can use the SNMP data for various purposes. *mrtg* itself is a graphing program that you will probably need to install. All major Linux distributions call the package *mrtg*. You can check if the program exists by typing:

```
# which mrtg
```

For now, let's setup a basic bandwidth graph available through your Web server. I'll be using */var/www* as the document root and */etc/mrtg.cfg* as the preferred location for the *mrtg* configuration file in the examples here. You'll need to create a */var/www/mrtg* directory, which will then be accessed as http://yourserver/mrtg/ to access the *mrtg* graphs.

If you don't have one already, create an *mrtg.cfg*, as shown in the *Listing One*. The data that follows is for my own server at `dtype.org`. Substitute all the appropriate values for your own server, including a name for this *mrtg* block config (mine is `verio228`). Multiple configs such as these can exist in the same *mrtg.cfg* file.

With this file in place, you should be able to run the following:

```
# mrtg -c /etc/mrtg.cfg
```

The first two times you run this, *mrtg* will complain that it does not have previous values. This is normal, as *mrtg* works by comparing numbers

Listing One: A Basic mrtg.cfg File

```
WorkDir: /var/www/mrtg
WriteExpires: Yes

Title[^]: Traffic Analysis for dtype.org

Title[verio228]: Verio 207.20.109.228
PageTop[verio228]: <H1>dtype.org at Verio 207.20.109.228</H1>
Target[verio228]: /207.20.109.228:public@localhost
MaxBytes[verio228]: 1280000
XSize[verio228]: 600
YSize[verio228]: 300
Options[verio228]: bits
```

(from SNMP in this case) at regular intervals. The first time you run *mrtg* with this config, there will be no basis data for the interval. You should also see files appear in */var/ www/mrtg/*.

Pointing a browser at http://yourserver/mrtg should give a file listing, including a `.html` file that includes the graphs you're looking for. The numbers will be disturbingly low right now because you haven't given *mrtg* time to run and collect data. By default, *mrtg* needs to run on five-minute crons.

You can learn a lot more about *mrtg* at the homepage: http://ee-staff.ethz.ch/~oetiker/webtools/mrtg/mrtg.html.

mrtg and *snmpd* both have options far beyond setting up simple bandwidth graphs. They are both very powerful and can be very useful to system administrators looking to monitor and control boxes remotely.

I want to upgrade a kernel on a server running remotely, but I am afraid of rebooting the machine and having it fail on startup. Is there a way to guarantee it comes back up?

I wish. There are a few steps you can take, however, to try to minimize your headaches here.

I'll outline my own procedure for doing this, which I must stress is only one way. It is, however, something I've done several times with some reliability. I have to assume that you're somewhat familiar with how to upgrade a kernel, or else you really shouldn't be attempting to do so remotely.

It helps a lot if you have the old kernel source; you can use this to retrieve the configuration for the kernel you are now running. The important file to find is the *.config* file for the running kernel.

1) Prep the new kernel: Download the new kernel source. It is a good practice to name the kernel source directory with a version number, so assuming you are upgrading to the 2.4.2 kernel, make sure there is no existing */usr/src/linux* (by moving it elsewhere or deleting the symlink) and then execute the commands found in *Listing Two.*

Listing Two: Getting the New Kernel Source

```
# cd /usr/src/
# wget http://ftp.sourceforge.net/pub/linux/kernel/v2.4/linux-2.4.2.tar.bz2
# cat linux-2.4.2.tar.bz2 | bunzip2 | tar xv
# mv linux linux-2.4.2
# ln -s linux-2.4.2 linux
# cp /usr/src/oldlinux/.config /usr/src/linux/
# cd /usr/src/linux/
```

If you were able to get an existing *.config* file from the */usr/src/linux/* directory, then execute:

```
# make oldconfig
```

This will only ask about configuration items that have changed since your last config. If your kernel versions are similar, this may be a small list, otherwise, it may take a while.

Regardless of whether or not you had the old config, you should really now check the config for your machine options. If you didn't have an old config, you'll need to do this from scratch. I like *menuconfig.*

```
# make menuconfig
```

It is *very* important to verify at least the basic hardware configuration of the box. I can't emphasize enough the need to carefully check configurations for hard drive types and drivers (SCSI and IDE), Ethernet adaptors, and other hardware your machine will need to boot and get onto the network.

Keep in mind that you will not be able to compile as a module something that you need to mount the root partition, because modules will not be available until the root partition is mounted.

2) Make and install kernel modules: Usually when I mess up a remote reboot it is because I forgot to install the new modules.

3) Check network setup: Be sure that your box defaults to its current network setup. This sounds redundant, but if you configured the box locally via an *ifconfig* line and didn't change the default interface behavior, you'll be out of luck if you don't check this.

While playing with network configs, keep in mind that you're currently logged into the box via the network. An `ifdown eth0` command will likely leave you stranded.

4) Check basic network daemons: If you are logging in through *ssh,* make sure that this starts in the init scripts.

5) Edit lilo.conf: Be sure to leave your existing kernel in the options as the default. After running *lilo,* your current kernel will still be the default. At this point, run:

```
# lilo -R newlinuxname
```

This will set *lilo* to boot *newlinuxname* (change as appropriate) for the next boot, but will still default to the old kernel for subsequent boots. This is very helpful if you hang the box, because then you can simply get an onsite person to power cycle it and boot into the known working kernel.

After a successful boot, don't forget to change the default back!

6) Reboot.

Good luck. No matter how many times I do this, I'm always anxious until the ping finally returns.

Drew Streib is a coder, admin, and writer with VA Linux Systems. He can be reached at tech@linux-mag.com.

Zonker's Product Picks

By Joe "Zonker" Brockmeier

TZO.com: At Your Service

Broadband access in the home is becoming more and more common. With all that bandwidth to spare, many folks would like to run a Web server from home. Unfortunately, it's kind of hard for people to surf to your site when you don't have a static IP, which many ISPs charge extra for.

TZO.com has a number of solutions for folks with a fat pipe and a dynamic IP. TZO.com provides client software that sends your assigned dynamic IP to TZO.com, where they handle the DNS service and point visitors to your Web site. Their basic plan allows you a subdomain like "myhost.tzo.com" for $24.95 a year, and the premier plan allows you a regular domain for $59.95 a year. They also offer mail storage and forwarding and other services for additional fees. For homes and small offices with modest Web needs, TZO.com might be the answer. You can download Linux, MacOS, Windows, and Java clients for the service from http://www.tzo.com.

Net::FTPServer is a New Way to FTP

If your FTP needs are a bit exotic, the Net::FTPServer might be the FTP server for you.

The Net::FTPServer project is a Perl module that has been tested under Linux 2.2 and Perl 5.00503, but should work on most Unix-type systems. In addition to standard FTP services, Net::FTPServer allows for the creation of a "virtual filesystem," as well as the serving of files from either a relational database or from memory.

Net::FTPServer is a project commissioned by Bibliotech. Code is available under the GNU General Public License from the project home at http://ftpserver.annexia.org/.

Nautilus: The Shell Game

The folks at Eazel have upped the ante in the battle for the computer desktop. Instead of playing catch-up with other OSes, the Nautilus file manager brings features to Linux not found anywhere else.

This slick GUI shell lets you view contents of files without opening them and zooms in and out on your folders, depending on how much detail you want to see. Nautilus is designed to browse local files and Web content.

The price for Nautilus is right too. Eazel is distributing Nautilus for free under the GPL and hoping to make money by offering Nautilus users services like online storage and software updates. Eazel was founded by former Apple developers Mike Boich and Andy Hertzfeld.

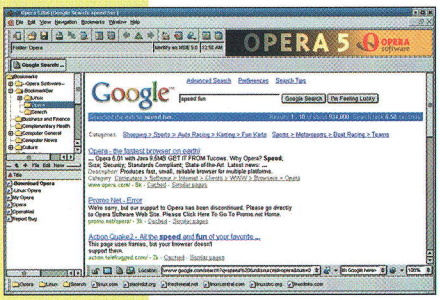
http://www.eazel.com

Innovator 8LX: The Penguin Speaks

Small businesses looking for a Linux voice processing solution should take a look at the Innovator 8LX. The 8LX is 1stCTI's Linux addition to their line of communications hardware that handles voice messaging, faxes, e-mail, and text messages.

The Innovator 8LX comes in two to eight port configurations. The Innovator has an easy to use GUI configuration system that can be used over a LAN or WAN. The Innovator works with 1stCTI's Windows-based desktop messaging and communications management software to provide a full communications solution.

The Innovator is expected to be available for shipment in July; prices are currently unavailable. 1stCTI offers a number of hardware and software communications solutions; they can be found at http://www.1CTI.com/.

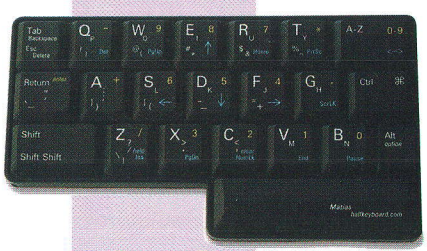


Opera for Linux: Small and Speedy

Browsing fanatics now have another option for surfing the Web under Linux — the Opera Browser. Opera for Linux has been under development for quite some time, but should be out of beta by the time you read this. Opera is very fast and has a different layout than Netscape, which some users prefer. The browser supports most Web standards, including standard HTML, CSS 1 and 2, XML, WML, SSL, and most JavaScript. Opera requires the Qt 2.2 libraries, which you already have if you're running KDE2. If not, you can just download the statically linked version.

The Opera Browser for Linux comes in two options. The free version is a Web browser with a "standard" sized ad banner at the top of the window. The $39 version is ad-free. The download is pretty slim, and packages are available in RPM, tar-gzip, and Debian formats. You can find Opera Software on the Web at http://www.opera.com/.

Half Keyboard: Half the Keyboard, Twice the Productivity



We may never know the sound of one hand clapping, but now we can find out what one hand typing sounds like. According to information on the Half Keyboard site, many users can achieve 62 words per minute with the Half Keyboard. Hunt and peck typists won't gain any speed, but they won't lose any either.

The Half Keyboard is a great solution if you're looking for a portable keyboard for your Palm Pilot or if you're putting together a wearable computer. The Half Keyboard comes in several models for Palm Pilots, Macs, and PCs. There are a couple of different Linux drivers available for the Half Keyboard as well.

The Palm Pilot, PS/2, and USB versions are priced at $99, and the portable version is priced at $199. You can find out more information about the Half Keyboard on their Web site: http://www.halfkeyboard.com/.

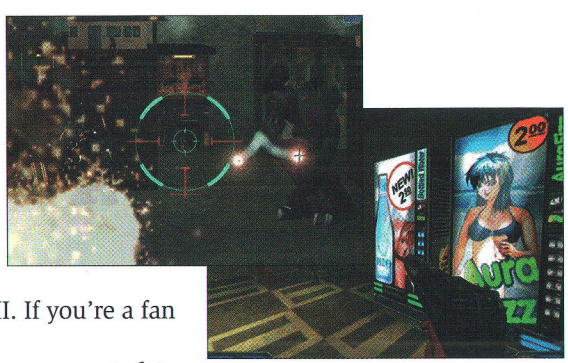
Macromedia Flash 5: Get Flashed!

It seems that almost every site contains Flash animation of some sort these days. Linux users have often been out of luck, because Flash support under Linux has been outdated at best.

It is now easier for Linux and Solaris users to waste more time on the Web with the Macromedia Flash 5 player. The player allows Linux and Solaris users to watch Flash multimedia presentations and animation.

Macromedia produces a number of multimedia players and content-producing tools for the Web. The Macromedia Flash Player is a free download available from their site: http://www.macromedia.com/shockwave/download/alternates/.

Shogo: Mobile Armor Division



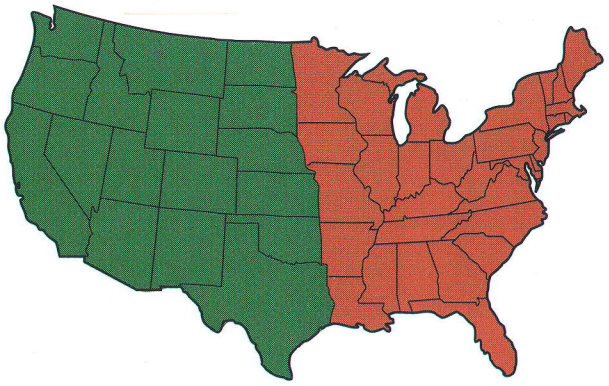
There's no better way to unwind after a hard day's work than to pilot a 10 meter tall robot that blows up other robots. Until you can do that, there's Shogo: Mobil Armor Division.

Shogo features heavy-duty 3D action powered by the LithTech 3D engine. Shogo offers some of the most realistic gore and lighting this side of Quake II. If you're a fan of first-person shooter action, Shogo should be on your wish list.

You can order Shogo: Mobile Armor Division from TuxGames.com. Shogo was ported to Linux by Hyperion Entertainment; they also port games to the MacOS and Amiga. Hyperion is based in Leuven, Belgium. http://www.hyperion-software.com/_linux/shogo_game.html

Got a product pick of your own for Zonker? Drop him a line at zonker@linux-mag.com.

Advertiser Index



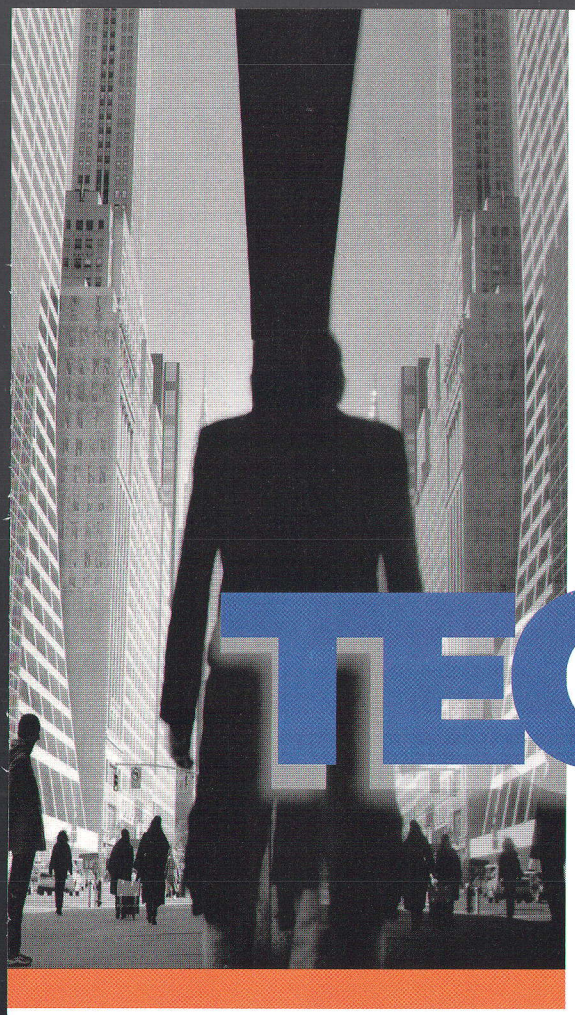
To advertise in *Linux Magazine*, please contact the appropriate ad sales rep listed below for a 2001 media kit containing an editorial schedule, rate card, and ad close dates.

East Coast Sales
Eric Derby • 973.786.0340 • eric@linux-mag.com

West Coast Sales
Mike Schaps • 650.966.8476 • mike@linux-mag.com

Linux Magazine is published monthly by InfoStrada LLC at 234 Escuela Avenue #64 Mountain View, CA 94040. The U.S. subscription rate is $29.95 for 12 issues. In Canada and Mexico, a one-year subscription is $39.95 US. In all other countries, the annual rate is $49.95 US. Non-US subscriptions must be pre-paid in US funds drawn on a US bank. Application to Mail at Periodicals Postage Rates is Pending at Mountain View, CA and Liberty, MO. POSTMASTER: Send address changes to *Linux Magazine*, P.O. Box 55731, Boulder, CO 80323-5731.

Article submissions and letters should be e-mailed to editors@linux-mag.com. *Linux Magazine* reserves the right to edit all submissions and assumes no responsibility for unsolicited material. Subscription requests should be e-mailed to linuxmag@neodata.com or visit our Web site at www.linux-mag.com.



This June, the leading companies and conferences in technology take center stage in New York City.

TECHXNY

Limited Engagement. One Week Only.

TECHXNY's full week of conferences, education summits and special events has something for just about everyone. The IT Innovation Conference focusing on innovative security solutions... an IT infrastructure built for speed...top-notch information-sharing architecture...Best of Brainshare...SAN and Clustering Summits... and the wide, wide, wireless world.

Marketing Integration Xchange, a conference and exposition, focusing on the use of technology to maximize marketing effectiveness and optimize customer relationships.

There's the **Finance Exchange**. The **Angel Society Forum** for entrepreneurs. **First Look** for senior-level execs. **Solutions Integrator Think Tech**. **PC Career Expo**. And still more targeted programs and events are on their way to bring you the cutting-edge of IT today.

To learn all about what there is to see, do, hear, and learn at TECHXNY this June, log on to **www.techxny.com**. And while you're there, make sure to register right away!

At PC EXPO, the expo portion of TECHXNY, you'll find mobile & wireless, business solutions, leading-edge Internet security advances, ASPs, the latest in Linux, storage technologies and more. The best products, services and real-world solutions—all delivered by the best vendors and solutions providers in the IT industry.

Conference Dates: June 25-28, 2001
Exposition Dates: June 26-28, 2001
Location: Javits Center, New York City

Flagship Sponsors

DOCULABS · The New York Times (Expect the World®) · Novell. · VERITAS

Platinum Sponsors

COMPAQ · handspring · MWT Mirror Worlds Technologies

Keynote Sponsor

BusinessWeek | www.businessweek.com

Sponsor of IT Innovation Conference

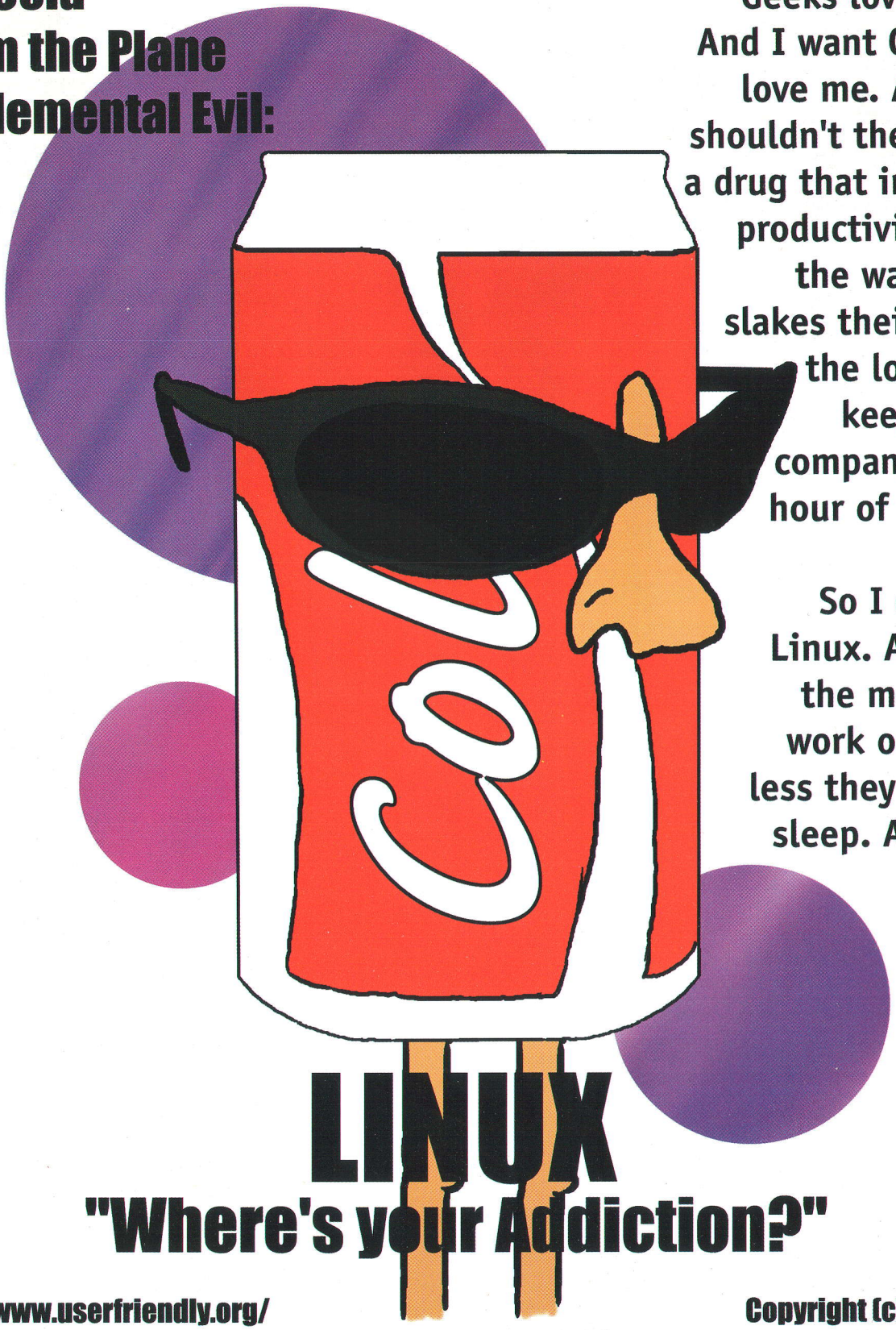
Forbes

REGISTER TODAY AT: www.techxny.com **Use source code: M3AA**

PC xpo · MIX Marketing Integration Xchange · TECHXNY · CMP

SHUTDOWN

Mr. Cola from the Plane of Elemental Evil:



Geeks love Linux. And I want Geeks to love me. And why shouldn't they? I am a drug that increases productivity. I am the water that slakes their thirst, the lover that keeps them company at any hour of the day.

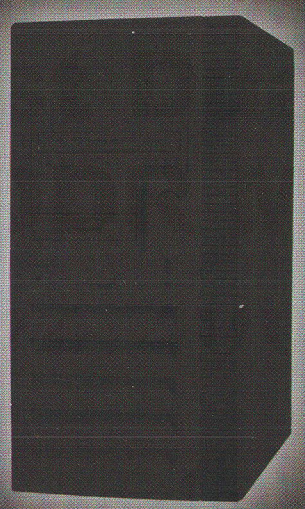
So I embrace Linux. After all, the more they work on it, the less they want to sleep. And then they're MINE.

LINUX

"Where's your Addiction?"

http://www.userfriendly.org/

The First Dual Athlon® With DDR Memory



Q2 2001

www.aslab.com

asl
ReDefining Linux Performance

Our embedded solutions support more platforms than a `70s disco.

Red Hat® Embedded Solutions help you better develop, deploy, and manage your products. No matter which platform you choose.

Red Hat has GNUPro tools and embedded operating systems to build a range of embedded products—from MP3 players to Mars Landers.

And you pick the platform. We support ARM/Thumb/StrongARM/XScale, Fujitsu FR, Hitachi SuperH, IA-64, IDT MIPS, MIPS32 4Kc, Motorola 68K, Matsushita AM32/AM33, NEC VR MIPS, PowerPC, Toshiba MIPS TX series, and x86/Pentium to name a few.

Get GNUPro, Embedded Linux, eCos, and our other Red Hat embedded solutions. They'll help speed your product to market and keep it from becoming tomorrow's eight-track.

Learn more: www.redhat.com/embedded.
Or contact us at embedded-info@redhat.com.

redhat.